

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

LEONARDO DE ALMEIDA COSTA

**AVALIANDO O ALGORITMO SPRAY-AND-WAIT NA
COMUNICAÇÃO ENTRE DISPOSITIVOS MÓVEIS ATRAVÉS
DE DTNS**

**ITABAIANA
2015**

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

LEONARDO DE ALMEIDA COSTA

**AVALIANDO O ALGORITMO SPRAY-AND-WAIT NA
COMUNICAÇÃO ENTRE DISPOSITIVOS MÓVEIS ATRAVÉS
DE DTNS**

Trabalho de Conclusão de Curso
submetido ao Departamento de
Sistemas de Informação da
Universidade Federal de Sergipe
como requisito parcial para a
obtenção do título de Bacharel em
Sistemas de Informação.

Orientador: MSC. JOSÉ AÉLIO DE OLIVEIRA JÚNIOR

**ITABAIANA
2015**

COSTA, Leonardo de Almeida.

Avaliando o Algoritmo Spray-and-Wait na Comunicação entre Dispositivos Móveis através de DTNs / Leonardo de Almeida Costa – Itabaiana: UFS, Ano. 66f.

Trabalho de Conclusão de Curso em Bacharel em Sistemas de Informação – Universidade Federal de Sergipe, Curso de Sistemas de Informação, 2015.

1. Delay and Disruption Tolerant Networks – DTNs.
 2. Redes de Computadores.
 3. Sistemas de Informação.
- I. Avaliando o Algoritmo Spray-and-Wait na Comunicação entre Dispositivos Móveis através de DTNs.

LEONARDO DE ALMEIDA COSTA

**AVALIANDO O ALGORITMO SPRAY-AND-WAIT NA
COMUNICAÇÃO ENTRE DISPOSITIVOS MÓVEIS ATRAVÉS
DE DTNS**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Sistemas de Informação da Universidade Federal de Sergipe (DSIITA/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Itabaiana, 24 de fevereiro de 2015

BANCA EXAMINADORA:

Prof(a) José Aélío de Oliveira Júnior, Mestre
Orientador
DSIITA/UFS

Prof(a) Methanias Colaço Rodrigues Júnior, Doutor
DSIITA/UFS

Prof(a) André Vinícius Rodrigues Passos Nascimento, Mestre
DSIITA/UFS

Dedico

Aos meus pais.

AGRADECIMENTOS

Em primeiro lugar ao verdadeiro Deus Jeová, por toda força, sabedoria e discernimento que me concedeu nos momentos mais difíceis.

Agradeço ao meus queridos pais, Mauricio e Neusa, por toda instrução moral e espiritual passados ao longo da minha vida, além do carinho, apoio e incentivo que me deram durante todo o período do curso.

A todos os meus familiares e amigos pelo apoio incondicional durante todo tempo dedicado à este trabalho. Agradeço imensamente o apoio inestimável do Sr. e Sra. Máximo que foram meu alicerce no início desta jornada. Sou muito grato também à família Rezende e à família de Tita, que me adotaram ao longo desses anos e me deram mais do que o apoio necessário. E agradeço ao meus queridos colegas que me apoiaram neste enorme desafio e foram meus parceiros durante muitos anos. Em especial, agradeço a Camila, Sthefanie e Charles, os quais se tornaram mais do que parceiros nesta luta e foram importantes em cada momento.

Enfim, agradeço ao meu orientador, Aélcio Júnior, por todo incentivo, apoio, paciência e conselhos para realização deste trabalho. E agradeço aos professores do departamento que se mostraram excelentes instrutores e exemplos de profissionais a serem seguidos de perto pelo resto da vida.

Epígrafe
“No meio de toda dificuldade encontra-se a oportunidade.”
(Albert Einstein)

COSTA, Leonardo de Almeida. **Avaliando o Algoritmo Spray-and-Wait na Comunicação entre Dispositivos Móveis através de DTNs**. 2015. Trabalho de Conclusão de Curso – Curso de Sistemas de Informação, Departamento de Sistemas de Informação, Universidade Federal de Sergipe, Itabaiana, 2015.

RESUMO

As redes tolerantes a atrasos e desconexões (Delay Tolerant Networks - DTNs) possuem características desafiadoras à comunicação nos seus diversos cenários. A sua principal característica é a comunicação intermitente devido às constantes desconexões e alta latência. Desta maneira, novos algoritmos de roteamento, com novas abordagens, foram desenvolvidos para que seja possível comunicar-se nesses ambientes, pois os algoritmos convencionais não atendem às necessidades destas novas redes. Entre estes, há o algoritmo de roteamento Spray and Wait, o qual foi avaliado neste trabalho sob diversos cenários que foram simulados. Estes cenários foram projetados e configurados a fim de aproximar-se ao máximo do cotidiano da sociedade moderna. Tais cenários possuem características de comunicação das redes móveis. Os resultados obtidos por meio das simulações indicaram um excelente desempenho do algoritmo nas diversas métricas avaliadas.

Palavras-chave: Redes, DTN, Roteamento, Spray-and-Wait.

ABSTRACT

Delay Tolerant Networks (DTNs) has challenging features to communication in its various scenarios. Its main feature is the intermittent communication from constant disconnections and high latency. In this way, new routing algorithms, with new approaches, have been developed to be able to communicate in these environments, once as conventional algorithms do not meet the needs of these new networks. Among these, there is the routing algorithm Spray and Wait, which was evaluated in this work under various scenarios were simulated. These scenarios was designed and configured in order to approach the most everyday of modern society. Such scenarios have communication features of mobile networks. The results of the simulations showed an excellent performance of the algorithm in the various metrics evaluated.

Keywords: Networks, DTN, routing, Spray-and-Wait.

LISTA DE FIGURAS

Figura 1 – As configurações salto-a-salto de um sistema “store-and-forward”	24
Figura 2 – Repasse de uma mensagem feita pelo mecanismo store-and-foward	27
Figura 3 – A camada de agregação.....	27
Figura 4 – Exemplo da movimentação de um nó sob a influência do modelo de mobilidade RWP	30
Figura 5 – Arquitetura do simulador The ONE.....	36
Figura 6 – Interface Gráfica do simulador The ONE	37
Figura 7 – Detalhes de um nó específico no simulador ONE e sua a rota de movimentação	38
Figura 8 – Mapa da cidade de Helsink, Finlândia	39
Figura 9 – Interface da ferramenta OpenJUMP com as rotas e pontos de interesse da cidade de Helsink	45

LISTA DE GRÁFICOS

Gráfico 1 – Quantidade de mensagens entregues por simulação	46
Gráfico 2 – Probabilidade de entrega da mensagem por simulação	46
Gráfico 3 – Quantidade de transmissões bem sucedidas por simulação	48
Gráfico 4 – Quantidade de mensagens apagadas por simulação	48
Gráfico 5 – Latência Média em segundos por rodada de simulações.....	49
Gráfico 6 – Quantidade de mensagens entregues por simulação	50
Gráfico 7 – Probabilidade de Entrega da mensagem por simulação	50
Gráfico 8 – Transmissões bem sucedidas por simulação	51
Gráfico 9 – Quantidade de Mensagens Apagadas por simulação	51
Gráfico 10 – Latência Média em segundos por simulação.....	51
Gráfico 11 – Quantidade de mensagens entregues por simulação	52
Gráfico 12 – Probabilidade de Entrega da mensagem por simulação	52
Gráfico 13 – Transmissões bem sucedidas por simulação	53
Gráfico 14 – Quantidade de Mensagens Apagadas por simulação	53
Gráfico 15 – Latência Média em segundos por simulação.....	53

LISTA DE TABELAS

Tabela 1 – Parâmetros gerais da simulação para o cenário urbano	41
Tabela 2 – Parâmetros gerais da simulação para o cenário veicular	43
Tabela 3 – Parâmetros gerais da simulação para o cenário emergência.....	45

LISTA DE ABREVIATURAS E SIGLAS

CONDOR – *Command and Control on the Move Network, Digital Over the Horizon Relay*

IEEE – Institute of Electrical and Electronics Engineers

SeNDT – *Sensor Networking with Delay Tolerance*

SISTAC/Bda – Sistema Tático de Comunicações de Brigada

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Motivação	17
1.1.1	Trabalhos relacionados	18
1.2	Objetivos.....	21
1.2.1	Geral.....	21
1.2.2	Específicos	21
1.3	Metodologia.....	21
1.4	Organização do trabalho	22
2	REDES TOLERANTES A ATRASOS E DESCONEXÕES.....	23
2.1	Definição	23
2.2	Características	24
2.3	Aplicações	25
2.4	Arquitetura	26
2.5	Modelos de Mobilidade	29
2.6	Roteamento	31
2.7	Spray and Wait	32
3	ANÁLISE QUANTITATIVA DE DESEMPENHO DO SPRAY AND WAIT.....	34
3.1	Ambiente de Simulação.....	34
3.1.1	Funcionamento e Características	34
3.1.1	Interface.....	36
3.2	Cenários e Parâmetros das Simulações	38
3.2.1	Cenários.....	39
3.2.1.1	Cenário Urbano	39
3.2.1.2	Cenário Veicular.....	42
3.2.1.3	Cenário de Emergência	43
3.3	Avaliação dos Resultados.....	46

4 CONCLUSÃO.....	54
3.1 Trabalhos Futuros	54
 REFERÊNCIAS	56
GLOSSÁRIO	58
 APÊNDICE	59
Apêndice A	59
Apêndice B	60
Apêndice C	62
Apêndice D	64
Apêndice E	66

1 INTRODUÇÃO

O crescimento do número de usuários utilizando dispositivos móveis inteligentes tem permitido que uma infinidade de aplicações com diversos propósitos sejam desenvolvidas para estes usuários. Ao mesmo tempo, novas necessidades de comunicação surgem para estes dispositivos que utilizam as tecnologias de comunicações sem fio como WiFi, WiMax, Bluetooth, entre outros. Hoje, produtos certificados que pertencem à classe de dispositivos de rede local sem fios (WLAN) baseados no padrão IEEE 802.11 (conhecido como WiFi) estão em 25 por cento dos lares em todo o mundo, e cerca de dois bilhões de dispositivos Wi-Fi foram vendidos em 2013 (ALLIANCE, 2014).

Entre as tecnologias de comunicações sem fio, existem as redes móveis sem fio, que são redes de computadores autônomas e auto organizáveis, cuja mobilidade dos nós altera suas topologias. Essas redes são compostas por dispositivos móveis que possuem alguma tecnologia de comunicação sem fio, e são utilizadas para criar uma rede *ad hoc*. Nessas redes, a topologia pode ser muito dinâmica, sem nenhuma infraestrutura pré-estabelecida, podendo ocorrer partições, fazendo com que os dispositivos não sejam capazes de se comunicarem uns com outros ou com um servidor central (OLIVEIRA et al. 2007) (ALMEIDA, 2007).

Em cenários com longos atrasos e frequentes desconexões, os protocolos TCP/IP não funcionam, desse modo protocolos alternativos são necessários. Redes com estas características específicas são denominadas Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* – DTNs) e um dos seus principais desafios é o roteamento, pois é preciso determinar rotas sem o estabelecimento de um caminho fim-a-fim (OLIVEIRA, 2008).

As redes DTNs são redes também formadas por nós móveis que se comunicam por meio de dispositivos sem fio, porém seus protocolos de roteamento não estabelecem um caminho fim a fim para permitir a comunicação e por este motivo suportam situações com conectividade intermitente e com longos atrasos, como os que ocorrem nas redes de emergência, nas redes veiculares e redes urbanas. Além destes casos, existem situações como: comunicações entre dispositivos com restrições de energia; comunicações rurais; comunicações submarinas; desastres ou catástrofes, onde a infraestrutura de redes com fio ou sem fio pode ter sido destruída ou pode não existir em locais remotos; entre outros (ALMEIDA, 2007) (OLIVEIRA, 2008).

Uma DTN pode ter melhor alcançabilidade especialmente em redes com nós esparsos devido às seguintes propriedades: comunicação baseada em mensagens assíncronas, a qual não necessita de um caminho fim a fim, pois as mensagens agrupadas podem ficar armazenadas nos nós até que seja estabelecida uma conexão; atrasos podem ser longos e variados; e tolerância a altas taxas de erros (ALMEIDA, 2007).

O desenvolvimento de produtos tecnológicos para estes cenários de DTNs ainda é escasso. Os protocolos de roteamento para DTNs podem servir como principais alternativas a fim de atender as necessidade de comunicações entre dispositivos fim a fim que se encontrem neste ambientes, onde a topologia dos nós é frequentemente alterada.

Entre os algoritmos da literatura existe o *Spray and Wait* (SPYROPOULOS et al. 2008) que tem como principal característica controlar o número de cópias de uma determinada mensagem na rede, sendo essa uma característica que tenta minimizar os efeitos de congestionamento de tráfego de redes causados principalmente pelos algoritmos epidêmicos.

Sendo assim, esse trabalho tem como finalidade realizar um estudo e avaliação sobre a utilização do *Spray and Wait* na comunicação entre dispositivos móveis nas seguintes redes: urbanas, veiculares e de emergência.

1.1 Motivação

A evolução tecnológica pode ser utilizada para melhorar a comunicação em cenários incomuns ou em ambientes onde a transmissão de dados seria valiosa para executar atividades ou operações específicas, promovendo assim ações com maior celeridade e precisão.

Em virtude da crescente disponibilidade de comunicação sem fio, tanto pessoas como carros e outros veículos de locomoção se tornam potenciais nós de criação, transmissão e recepção de dados. Todavia, para atender aos cenários que possuem uma infraestrutura dinâmica é necessário utilizar as abordagens dos roteamentos existentes em ambientes DTNs.

Diante das possibilidades que surgiram com as novas tecnologias, torna-se possível a criação de redes de comunicação em cenários como: rede veiculares, com objetivos de coletar informações de uma determinada estrada ou malha rodoviária, promover melhorias na segurança entre os veículos que se deslocam pela rodovia, controlar o tráfego para evitar congestionamentos, entre outros; redes em regiões de desastre, como terremotos, furacões, inundações, acidentes nucleares, entre outros, permitindo uma melhor coordenação e

articulação das equipes de resgate nas situações de emergência para executarem seus respectivos planos de ação com mais eficiência (ALMEIDA, 2007).

Com a possibilidade da criação dessas redes de comunicação se faz necessário analisar o desempenho que o algoritmo de roteamento *Spray and Wait* tem durante a transmissão dos dados entre os nós que estariam envolvidos nos cenários, de forma que sejam identificadas as características favoráveis para o uso do mesmo e melhorias para um melhor desempenho em cada situação.

1.1.1 Trabalhos Relacionados

O artigo de (BRANCO et al., 2010) apresentou um estudo que buscou avaliar o desempenho de uma rede MANET-DTN implementada para operar em um cenário típico de desastre ou emergência.

O cenário simulado foi composto por: um centro de comando e controle, que concentrava todas as informações e decisões; hospitais, que serviam como pontos de atendimento; e ambulâncias, que desempenhavam o papel de nós móveis e realizam as operações de socorro, resgate e transporte. O sistema de comunicações MANET-DTN ficava embarcado nessa infraestrutura. As simulações foram feitas com base em três bairros do Rio de Janeiro, sendo estes: Botafogo, Méier e Ilha do Governador.

Para avaliar os protocolos de roteamento, foi utilizado o simulador The ONE. Todos os nós móveis utilizaram o padrão de mobilidade *Shortest Path Map Based Movement* e suas velocidades foram ajustadas para o tráfego “em situação de desastre”, ou seja, compatível com o tipo de pavimentação irregular passível de ser encontrada em uma situação de desastre e a malha viária da região. Após a análise dos dados gerados pelas simulações, observou-se que os protocolos replicadores, epidêmico e profeta, tiveram um desempenho superior em relação aos outros protocolos comparados no trabalho.

Em (CRUZ & ASSAKURA, 2010), foi realizado uma análise sobre a influência das métricas sociais no desempenho de redes tolerantes a atraso e desconexões. Para realizar esta análise foram utilizados mais três protocolos de roteamento em DTN: um de replicação mais tradicional, o Epidêmico, que usa múltiplas cópias de mensagens para o encaminhamento; o *Spray and Wait*, que diminui o número de cópias a ser enviado e guarda parte das mensagens em cada nó; e o *BubbleRap* sendo este um protocolo que utiliza métricas sociais no seu algoritmo de encaminhamento. Além disso, foi criado um cenário com base numa área real do

centro da cidade do Rio de Janeiro. Ao fim do trabalho, observou-se que o *Spray and Wait* e o *BubbleRap* foi o único que não apresentou mudanças no comportamento das métricas de desempenho com a variação no tamanho do buffer, nem mesmo com o aumento da carga dos dados. Além disso, o *BubbleRap* gerou pouca sobrecarga na rede, apresentou uma taxa de entrega mais próxima do epidêmico do que o *Spray and Wait*.

Em (JÚNIOR, 2001), foi realizado uma análise do tráfego de dados em redes *bluetooth*, a qual estudou o desempenho da descoberta de dispositivo Bluetooth; como ocorria o estabelecimento das conexões, bem como a negociação QoS. Vários modelos de tráfego para dados foram desenvolvidos e analisados via simulação; e foi analisado o desempenho do Bluetooth sujeito a este tráfego (em termos de qualidade de serviço – QoS). Como a tecnologia ainda estava surgindo, conclui-se que muitas pesquisas e melhoramentos ainda deviam ser feitos, e argumentou-se que muitos ajustes futuros precisariam ser feitos a fim de aproximar as simulações da realidade.

Ainda em (MELO, 2011), foi avaliado o comportamento e viabilidade de implantação de uma rede tolerante a atrasos ou desconexões que utilizasse a infraestrutura do transporte aéreo, no âmbito do território brasileiro. Para isso, foi caracterizada a DTN dentro do escopo da estrutura do tráfego aeroviário nacional, levando em consideração seus componentes, limites e mobilidade; apresentando-se as simplificações realizadas para modelar a troca de dados dentro da DTN proposta; implementando-se um simulador de troca de mensagens em DTNs usando tráfego aéreo nacional; avaliando-se o tempo de atraso para entrega de mensagens desde sua geração no nó de origem até a entrega no nó destino; analisando-se o número de saltos que as mensagens precisaram dar entre os nós da rede até chegarem ao seu destino; e também a viabilidade de utilização desta DTN em comparação com a Internet. Por fim, na análise dos dados verificou-se um atraso elevado para entrega de mensagens de pequenos tamanhos, o que tornaria inviável o uso prático para este caso. Além disso, observou-se que o número de saltos médios ficou entre três e quatro até a mensagem ser entregue e que a probabilidade de entrega da mensagem chegou a quase 100%.

Em (ALMEIDA, 2011), foi apresentado um estudo do desempenho dos principais algoritmos MANET e DTN existentes na literatura, o qual avaliou não apenas métricas de entrega e atraso, mas também o consumo de energia e o descarte das mensagens. Foram identificadas as principais características das classes que maximizam o desempenho geral da rede em diversos cenários, bem como quais seriam as possíveis combinações, dentre os modelos de mobilidade avaliados, entre as características que permitiriam o melhor

desempenho. Desse modo, observou-se que o desempenho de cada protocolo dependeu do cenário em que este foi analisado, ou seja, nenhum deles apresentou um desempenho superior em todos os cenários. E confirmou-se que os protocolos DTNs entregam mais mensagens quando a densidade de nós é baixa e a mobilidade é alta. Por outro lado, observou-se que, quando a densidade é alta e a mobilidade é baixa, os protocolos DTN apresentam alto consumo de energia e longos atrasos.

Além de simulações de diversos cenários, sendo estes realistas ou modelados, para análise de desempenho dos algoritmos mais conhecidos, observam-se também trabalhos como o de (FERNANDES, 2009), o qual propôs um mecanismo de segurança para DTNs que segue as três premissas básicas de segurança de redes (autenticidade, confidencialidade e integridade). Além disso, foi feita uma investigação do impacto do uso da segurança proposta em diferentes cenários e foi realizada uma avaliação do mecanismo de segurança sob a influência de diferentes protocolos de roteamento. Por fim, com a análise dos resultados observou-se que o mecanismo de segurança não causou uma grande influência no desempenho da rede quando comparado com o desempenho da rede sem o mecanismo de segurança.

Outra proposta notável foi feita por (JOB, 2010), a qual propôs uma alteração na camada de agregação da arquitetura das DTNs por empregar a arquitetura de correio eletrônico para criar uma plataforma de comunicação a fim de executar as principais funções desta camada. O funcionamento da arquitetura é baseado no conceito de contatos oportunistas, os quais são contatos que acontecem através de encontros entre os nós que não foram programados, criando assim, oportunidades de comunicação entre os demais nós da rede. Com realização de experimentos, foi possível obter resultados satisfatórios, permitindo verificar a viabilidade da proposta.

Por fim, em (JUNIOR, 2014), salientou-se sobre as diversas aplicações que poderiam ser feitas por usar *anycasting* em DTNs, como por exemplo, uma mensagem de emergência poderia ser enviada por um veículo (um nó) envolvido em um acidente a fim obter ajuda das equipes de socorro, seja de um caminhão de bombeiros ou de uma ambulância. Neste caso, qualquer um dos veículos, ou dos nós do grupo, seria capaz de processar a mensagem (e, possivelmente, poderia encaminhá-la para outros serviços, caso necessário). Com base nisso, foi proposto um algoritmo baseado no conceito da força de atração de um campo magnético de uma mensagem, de forma a guiá-la para um dos destinos em potencial. Uma força de atração é atribuída a cada nó. Quanto mais forte a força de

atração, seria mais provável uma mensagem ser encaminhada para esse nó. Esta propriedade é transitiva, o que significa que um nó pode indiretamente exercer influência magnética sobre outros nós vizinhos. Como o trabalho ainda está em desenvolvimento, deve-se aguardar os resultados da implementação do algoritmo e a execução em simulações para a coleta e análise das métricas.

1.2 Objetivos

1.2.1 Geral

Este trabalho tem como objetivo principal a avaliação do desempenho do algoritmo *Spray and Wait*, para comunicação entre dispositivos móveis nas seguintes redes: urbanas, veiculares e de emergência.

1.2.2 Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram realizados:

- Simulações das redes urbanas, veiculares e de emergência utilizando o algoritmo *Spray and Wait* no simulador The ONE.
- Análise dos resultados obtidos;
- Fomento à pesquisa da área de DTNs e;
- Propor aplicações futuras que utilizem o trabalho desenvolvido como base.

1.3 Metodologia

Ao longo deste trabalho foi feito um levantamento bibliográfico sobre as redes tolerantes a atrasos e desconexões, seus fundamentos, características, arquitetura e roteamento das mensagens entre os nós de uma rede. Em seguida, foi realizado o estudo sobre o algoritmo de roteamento *Spray and Wait*.

Na etapa seguinte, foi realizado o estudo sobre o simulador The ONE com a finalidade de compreender o seu funcionamento; como alterar suas configurações e definir o algoritmo alvo do estudo para implementar os cenários que foram simulados posteriormente;

bem como gerar os relatórios necessários para a avaliação do seu desempenho em cada cenário.

Visto que a DTN tem a infraestrutura dinâmica como sua principal característica, se fez necessário fazer um estudo sobre os modelos de mobilidade dos nós, pois em alguns casos estes nós podem representar a movimentação humana ou outros mamíferos. Desse modo, foi importante definir um modelo que tivesse características semelhantes a movimentação supracitada.

Em seguida, as configurações dos cenários foram definidas e as simulações foram executadas. Cada simulação gerou relatórios que foram previamente escolhidos e definidos nas configurações. Por fim, foi feita a análise dos relatórios.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma. O capítulo 2 fornece uma visão geral sobre as redes tolerantes a atrasos e desconexões e, em seguida, apresenta a definição, as características, as aplicações, a arquitetura, os modelos de mobilidade e a importância deles numa DTN, articula de um modo geral sobre os principais algoritmos de roteamento, depois aprofunda-se no algoritmo de roteamento *Spray and Wait*.

O capítulo 3 faz a análise quantitativa de desempenho do *Spray and Wait*. No entanto, ele começa descrevendo qual foi o ambiente de simulação escolhido, o funcionamento e as características do simulador e a sua interface. Em seguida, são descritos os cenários e os parâmetros definidos para executar as simulações. Posteriormente é feita a avaliação dos resultados dos relatórios gerados pelo simulador. O capítulo 4 faz uma consideração sobre os trabalhos futuros e por fim, no capítulo 5 é feita a conclusão.

2 Redes Tolerantes a Atrasos e Desconexões

2.1 Definição

Nos últimos anos, o aumento da produção de dispositivos móveis de comunicação, tais como, telefones celulares, assistente pessoal digital (PDAs), computadores portáteis como notebook ou netbook e outros estão contribuindo para o surgimento de novos ambientes de redes. Estes novos ambientes utilizam comunicações sem fio, comunicações entre dispositivos móveis, comunicações entre dispositivos com restrições de energia, comunicações rurais, comunicações submarinas e, até mesmo comunicações interplanetárias (OLIVEIRA, 2008) (JOB, 2010).

As redes formadas por esse tipo de dispositivo apresentam características como grande mobilidade, baixa autonomia de baterias, pouca ou nenhuma infraestrutura, dificuldade de manter uma comunicação fim-a-fim com baixa latência e baixa perda de pacotes. Tais fatores colaboram para o aparecimento de cenários desafiadores, onde os atrasos e as desconexões ocasionam o funcionamento inadequado da rede (OLIVEIRA, 2008) (JOB, 2010).

Devido a estas características, as redes que consideram estes aspectos foram denominadas Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks – DTNs*) e, mais recentemente receberam outras definições e títulos como: redes desafiadoras (*CHALLENGED NeTworkS – CHANTS*) (ALMEIDA, 2011 apud CHEN & MURPHY, 2001); Redes Oportunistas (*Opportunistic Networks*) (ALMEIDA, 2011 apud RAMANATHAN et al., 2007); Redes com conectividade intermitente (*Intermittently Connected Networks – ICN*) (ALMEIDA, 2011 apud GHOSH et al., 2006); ou ainda, Redes com Desafios de Conectividade (*Connectivity-Challenged Networks*) (ALMEIDA, 2011).

Uma DTN pode chegar a ter atrasos da ordem de horas e, até mesmo dias. A variação do atraso também pode chegar a estes valores. Em relação às desconexões, estas podem ocorrer: pela alta mobilidade que provoca constantes mudanças na topologia da rede; por péssimas condições de comunicação; por economia de recursos como em sensores sem fio que dormem para poupar energia; por negação de serviço como o ato do inimigo sujar a frequência; entre outros. Estes eventos podem resultar em uma conectividade intermitente da

rede durante um período ou, ainda, pode ser que um caminho entre a origem e o destino nunca chegue a ficar completamente conectado (OLIVEIRA, 2008).

2.2 Características

Segundo (OLIVEIRA, 2008), as principais características encontradas na redes tolerantes a atrasos e desconexões são:

- Os atrasos longos e/ou variáveis; (Oliveira, 2008 apud JONES, 2005) afirma que o atraso fim a fim pode variar entre horas ou dias e que é determinado através da soma dos tempos de atraso e salto. O atraso se divide em: tempo de espera, que é tempo em que o nó aguarda por um nó intermediário ou nó destino para encaminhar as suas mensagens; atraso nas filas dos próprios nós antes de uma mensagem ser transmitida; atraso de transmissão; e atraso de propagação do sinal de latência entre dois nós.
- As frequentes desconexões, que podem provocadas pelas alterações na topologia da rede ou por situações desfavoráveis à comunicação, como péssimas condições ou economia de recursos.
- Ainda segundo (Medeiros et al, 2009), a capacidade de armazenamento (*store-and-forward*) é provavelmente a principal característica das DTNs. Estes tipos de redes possuem dispositivos onde a informação fica armazenada até o surgimento de um enlace com outro dispositivo cuja transmissão possibilite alcançar seu destinatário. Este processo de transmissão é exemplificado na Figura 1.

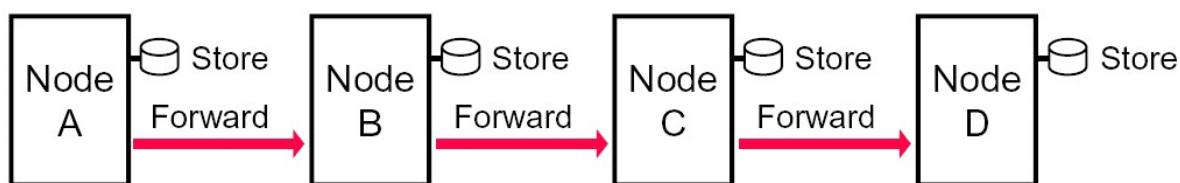


Figura 1 - As configurações salto-a-salto de um sistema “*store-and-forward*” (MEDEIROS et al., 2009 apud WARTHMAN et al., 2003)

2.3 Aplicações

Diversas localidades do mundo não possuem a infraestrutura necessária para a utilização de aplicações comuns como o correio eletrônico e a *World Wide Web* (WWW). Normalmente são regiões rurais ou regiões residenciais habitadas por pessoas de baixo poder aquisitivo. Essas localidades encontram-se, em geral, afastadas dos grandes centros, onde existem diversas formas de acesso à Internet como a banda larga e o modem discado (OLIVEIRA et al, 2007).

Existem diversos cenários para uso das redes DTN, entre eles se encontram as redes de sensores sem fio, onde os nós precisam economizar energia e por isso permanecem desligados periodicamente, causando o particionamento da rede e conectividade intermitente (OLIVEIRA et al, 2007).

Outro exemplo são as redes táticas militares, pois estão sujeitas a frequentes interrupções nas comunicações fim-a-fim, gerando assim, falta de confiabilidade, o que tornariam inviáveis as comunicações em ambientes de campos de batalha. Por isso, projetos militares como o CONDOR (*Command and Control on the Move Network, Digital Over the Horizon Relay*) e Sistema Tático de Comunicações de Brigada (SISTAC/Bda), usado pelo exército brasileiro, fazem uso das redes DTN para resolver os problemas de comunicação em seus ambientes (JOB, 2010).

O projeto SeNDT (*Sensor Networking with Delay Tolerance*) (ALMEIDA, 2011 apud PROJECT, 2005) fez aplicação da DTN para monitoramento ambiental, o qual auxilia as organizações públicas e privadas preocupadas com o meio ambiente a monitorar a qualidade da água, dos lagos e também os níveis de poluição sonora em rodovias. Para a transmissão dos dados foram utilizados nós móveis ou “mulas de dados” que trafegavam na região monitorada, desse modo havia a possibilidade de coleta de informações e conectividade.

Citando ainda outro projeto, temos o Zebranet (JUANG et al., 2002), que envolve pesquisadores da vida selvagem e sistemas computacionais visando o monitoramento, sem a intervenção de humanos, de uma área de milhares de quilômetros quadrados do parque nacional no Quênia. Colares rastreadores foram implantados em zebras e outros animais que convivem no mesmo ecossistema para monitorar o comportamento delas. Este equipamento registra informações detalhadas periodicamente. São registradas: estimativas de posicionamento, dados ambientais e do clima, os movimentos dos bandos e dados fundamentais entre as espécies. O objetivo é entregar, aos pesquisadores, informações

coletadas ao longo dos meses ou anos em que o sistema ficar em operação. Visto que nem sempre os colares estão próximos à estação base, os dados não podem ser enviados diretamente, assim é necessário usar os colares intermediários. No projeto, todos os nós são fontes de dados, enquanto a estação base apenas coleta os dados. Os nós são móveis e as estações base também são móveis, dependendo das rotas tomadas pelos pesquisadores com seus veículos.

Os desastres podem ainda ser citados como exemplos, onde as situações rompem o funcionamento normal da sociedade e a economia de uma região, e podem ter origens naturais, tecnológicas ou humanas. Catástrofes são desastres em grandes proporções que acarretam grandes prejuízos humanos e materiais. As situações ocasionadas por desastres naturais, tecnológicos ou causados pelo homem são chamadas de cenários de emergência. Os esforços para reduzir os danos físicos e sociais de um desastre exigem que várias organizações coordenem seus esforços. Os agentes em cenários de emergência, chamados de respondedores, podem ser agentes da defesa civil, médicos, bombeiros, policiais e voluntários (ALMEIDA, 2011).

2.4 Arquitetura

Para contornar os problemas de atrasos e desconexões, as DTNs se servem da técnica de comutação de mensagens além de armazenamento persistente dos dados. Quando uma mensagem precisa ser enviada, ela é armazenada e encaminhada de nó para nó, desde a origem até o destino. Por utilizar essa técnica, diz-se que as DTNs são redes do tipo “armazena-e-encaminha” (*store-and-forward*), ou seja, primeiro a mensagem é recebida integralmente e armazenada para, depois, ser enviada ao próximo nó, que pode ou não ser o destino, a figura 2 ilustra os processos da troca de mensagens. Como as DTNs não operam sobre enlaces que estão sempre disponíveis, é esperado que os nós armazenem mensagens durante algum tempo, sendo preciso alguma forma de armazenamento persistente e robusto para preservar as informações diante de reinicializações no sistema (OLIVEIRA, 2008 apud WARTHMAN, 2003).

Segundo (OLIVEIRA et al, 2007), a arquitetura DTN prevê a utilização da técnica de comutação de mensagens e o armazenamento persistente dos dados definindo uma sobre camada (*overlay*) abaixo da camada de aplicação. Esta nova camada é denominada “camada de agregação” (*BundleLayer*) e o protocolo de agregação é executado em todos os nós

pertencentes à rede DTN, denominados nós DTN, da origem até o destino, à semelhança da camada IP. As “subredes” são denominadas redes regionais e a arquitetura sobre a camada permite tornar a DTN totalmente independente das diversas redes regionais, permitindo que as aplicações se comuniquem através de múltiplas regiões. Para garantir interoperabilidade com qualquer tipo de rede, esta sobre camada se situa acima da camada transporte das redes que se servem do perfil de protocolos TCP/IP.

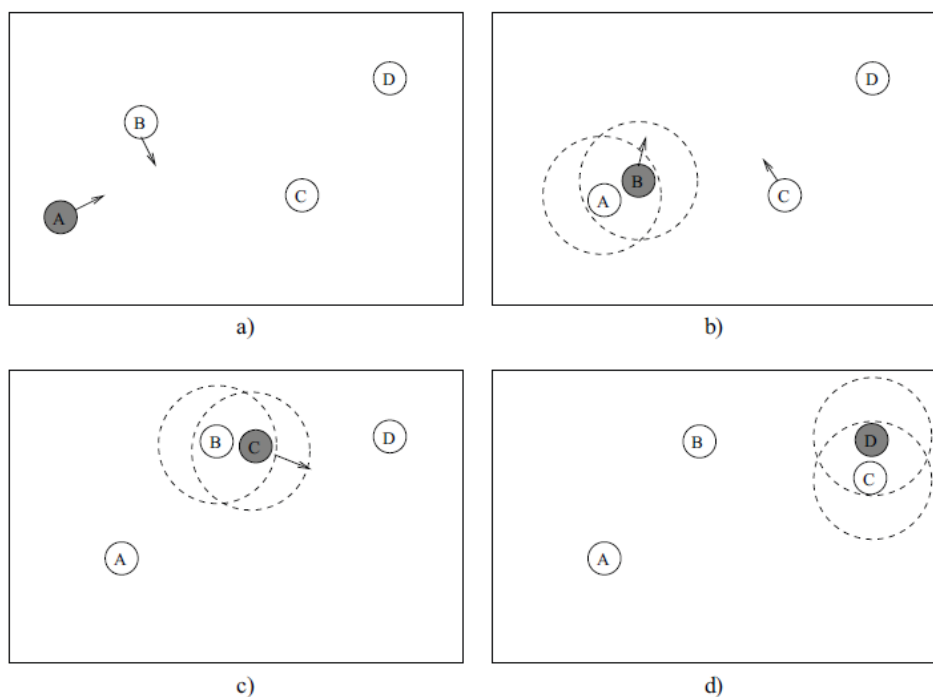


Figura 2 - Repasse de uma mensagem feita pelo mecanismo *store-and-forward* (NUNES, 2010)

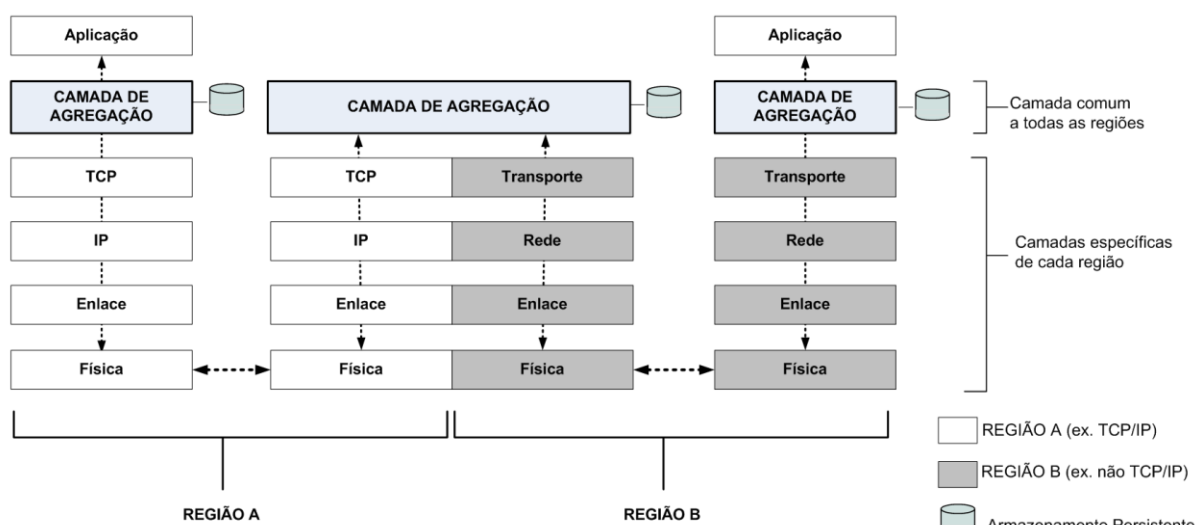


Figura 3 - A camada de agregação. (OLIVEIRA et al, 2007)

Como ilustrado na Figura 3, as camadas abaixo da camada de agregação são definidas de acordo com a conveniência do ambiente de comunicação de cada região, podendo ser específicas para cada região englobada pela DTN.

Em DTNs não é assumido que todos os nós são alcançáveis e podem ser contatados a qualquer instante. Essa característica das DTNs contrasta fortemente com o que é assumido para a internet convencional, a qual considera que as entidades comunicantes estão sempre alcançáveis. Por isso, é importante que deva ser considerado na arquitetura DTN o um conceito de contato. Um contato corresponde a uma ocasião favorável para os nós trocarem dados. A arquitetura DTN classifica os contatos em: persistente, sob demanda, programado, oportunista e previsível. Possibilidades de falhas sempre existem para qualquer tipo de contato. Porém, dependendo do tipo de contato, as falhas podem ser mais frequentes, como é o caso dos contatos previsíveis (OLIVEIRA et al, 2007).

Em DTNs, a topologia da rede é dinâmica, assim alguns enlaces podem existir durante apenas alguns intervalos de tempo e esse tempo durante o qual um enlace existe é chamado de contato. Além disso, os contatos podem ser determinísticos, previsíveis ou aleatórios (DUARTE, 2008).

Quando todos os contatos são determinísticos, diz-se que o cenário é determinístico, ou seja, tem-se conhecimento de quando ocorrem os contatos da rede ou, em última instância, da topologia da rede em qualquer instante de tempo. Ao contrário do cenário determinístico, no cenário estocástico (ou dinâmico) o comportamento da rede não é completamente conhecido (DUARTE, 2008).

Os cinco tipos de contatos são: os contatos persistentes, que são aqueles que estão sempre disponíveis; os contatos sob demanda, que são aqueles que requerem alguma ação para que sejam instanciados, mas que, uma vez acionados, funcionam como contatos persistentes até serem encerrados; os contatos programados, onde uma agenda de contato pode ser preestabelecida entre dois ou mais nós antes que ocorra a troca de informações, o horário e a duração de cada contato são estabelecidos previamente entre os nós comunicantes; os contatos previsíveis são aqueles nos quais os nós podem fazer previsões sobre o horário e a duração dos contatos com base em históricos de contatos previamente realizados; por fim, os contatos oportunistas, que ocorrem diante de encontros não previamente programados entre os nós (OLIVEIRA et al, 2007).

2.5 Modelos de Mobilidade

Visto que muitas das redes tolerantes a atrasos e desconexões possuem nós que representam seres humanos ou animais, muito se tem estudado a respeito da mobilidade dos nós, pois em cada caso os mesmos podem estar se deslocando em direções diferentes, em um determinado intervalo de tempo.

Segundo (FERNANDES, 2009), a mobilidade dos nós pode ser representada através de mobilidade sintética, onde modelos matemáticos determinam o movimento dos nós; ou através de dados reais, onde rastros (*traces*) de mobilidade real são utilizados para representar a mobilidade dos nós. Estes rastros de mobilidade real podem ser dados pela movimentação de pessoas em um determinado local, como um shopping center, deslocamento de carros em determinadas ruas de um centro urbano, ou ainda, a movimentação de animais na natureza, por exemplo.

Segundo (MOSCHETTO et al., 2011 apud RHEE, 2011) o modelo matemático *Levy Walk* permite capturar bem as mudanças das movimentações dos seres vivos, uma vez que estes realizam movimentos curtos em torno de pontos de interesse e alternam para movimentos longos usando o menor caminho para outros pontos de interesses distantes. No entanto, este modelo considera que a movimentação dos nós está livre de obstáculos, o que difere da movimentação em cidades, uma vez que o deslocamento acontece em ruas e avenidas.

Segundo (RASHID et al, 2011), vários modelos de mobilidade podem ser categorizados de acordo com suas dependências espaciais e temporais. A dependência espacial é uma medida de como dois nós são dependentes em seus movimentos. Se dois nós estão se movendo na mesma direção, então eles têm alta dependência espacial. A dependência temporal é uma medida de quanto da velocidade atual (magnitude e direção) estão relacionados com a velocidade anterior. Nós com mesma velocidade têm alta dependência temporal.

O modelo de mobilidade aleatória – *Random Waypoint Model* (RWP) – proposto por Johnson and Lee funciona movendo os nós aleatoriamente de acordo com a velocidade e direção em intervalos de tempo regulares. Neste modelo um nó móvel se mantém na posição por um determinado período de tempo, uma vez expirado o tempo, o nó move-se para o novo destino, escolhendo uma velocidade aleatória a partir de variação mínima e máxima pré-estabelecida, a figura 4 demonstra a movimentação com base neste modelo.

Há ainda o modelo *Randon Walk* (RW) que descreve o movimento relativo de um nó que se move da posição atual para um novo local por um intervalo de tempo t , escolhendo aleatoriamente uma direção e velocidade de intervalos pré-definidos. No final de cada intervalo de tempo uma nova direção e velocidade são calculados. (RASHID et al, 2011 apud CAMP et al, 2002).

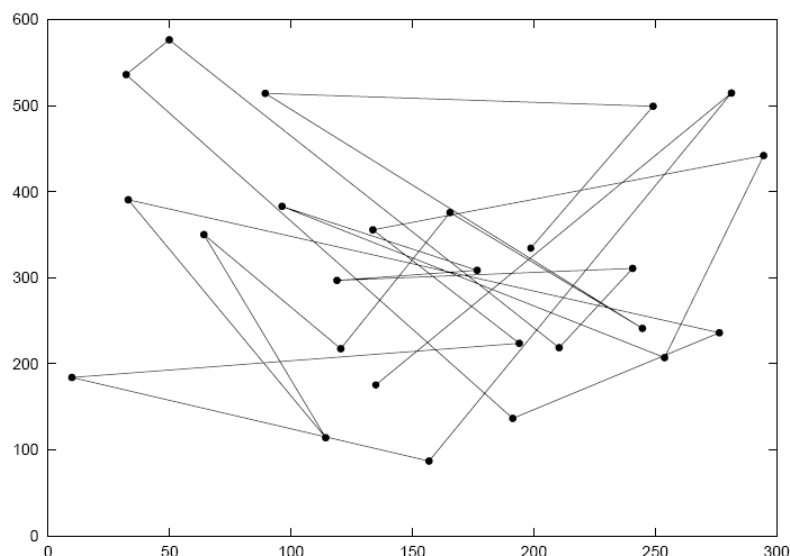


Figura 4 - Exemplo da movimentação de um nó sob a influência do modelo de mobilidade RWP (FERNANDES, 2009).

Ainda segundo (RASHID et al, 2011), nos modelos baseados em mapas simples aleatórios (*Map Mobility Models – MBM*), os nós se movem para posições determinadas aleatoriamente sobre o mapa, mas seguem as estradas definidas pelos dados do mapa. Há ainda outro, que é uma extensão do MBM, denominado como modelo de mobilidade de mapas de rotas – *Route Map Mobility Models (RMBM)* –, mas que é usado apenas para mobilidade de grupos de veículos (RASHID et al, 2011 apud KERÄNEN et al, 2007).

O modelo *Shortest Path Map-based Movement (SPMBM)* (MOSCHETTO et al., 2011 apud KERÄNEN et al., 2009) tem um comportamento semelhante ao *Levy Walk* no que diz respeito à seleção de menor caminho, pois ao invés de vagar aleatoriamente em um mapa, os nós fazem uso do algoritmo de Dijkstra para calcular o caminho mais curto, a partir do local corrente, para um destino escolhido aleatoriamente. Após chegar ao destino, o nó aguarda um intervalo de tempo t , então depois é direcionado para outro destino. (KERÄNEN et al, 2007)

2.6 Roteamento

Um desafio comum a todas as categorias de DTN é o roteamento, pois é preciso projetar protocolos capazes de superar os problemas dos atrasos extremamente longos e das frequentes desconexões, já que os protocolos convencionais não estão aptos a manipular eficientemente a transmissão de dados em DTNs (OLIVEIRA et al. 2007).

Como existem diversos protocolos de roteamento para DTNs que se diferenciam pela quantidade de informações a respeito da mobilidade dos nós que cada um utiliza, serão apresentados somente alguns dos principais protocolos de roteamento para DTNs.

Em (ALMEIDA, 2011 apud VAHDAT & BECKER, 2000) foi apresentado o algoritmo Epidêmico. Neste, as mensagens são repassadas para o maior número de nós possível, como em uma epidemia. Este algoritmo possui a menor latência e a maior taxa de entrega dentre os protocolos DTN, considerando um cenário ideal, onde não existem restrições de banda, energia e *buffer*. Este protocolo possui uma grande desvantagem, pois um grande número de agregados é trocado e por isto uma grande quantidade de recursos da rede é utilizada. Em cenários onde existem restrições de energia, banda ou *buffer* este algoritmo pode apresentar queda no seu desempenho.

A ideia é que a mobilidade dos nós na rede possibilite que eles entrem no alcance de transmissão uns dos outros periodicamente e de maneira aleatória. Logo, a mobilidade dos nós é utilizada como solução para a entrega de mensagens, ao invés de ser tratada como um problema que precisa ser superado na rede (OLIVEIRA et al. 2007).

O *Probabilistic Routing Protocol using History of Encounters and Transitivity* (PROPHET) (ALMEIDA, 2011 apud LINDGREN et al., 2003) é um algoritmo probabilístico de roteamento para redes DTN que explora a não aleatoriedade nos movimentos dos nós em uma rede, esperando assim melhorar a entrega de agregados mantendo baixo o consumo de recursos.

Assim como acontece no roteamento epidêmico, quando dois nós iniciam um contato, são trocadas as listas com informações que identificam as mensagens armazenadas em cada nó. A diferença é que agora existe uma informação extra para cada mensagem indicada na lista. Essa informação corresponde à probabilidade de cada nó entregar mensagens para um destino conhecido b ($P(a,b) \in [0,1]$). O valor de $P(a,b)$ aumenta sempre que a e b se encontram. Estratégias de descarte de mensagens baseadas na estimativa da

probabilidade de entrega também podem ser utilizadas para aumentar o desempenho da rede quando os nós possuem *buffers* limitados (OLIVEIRA et al. 2007 apud DAVIS et al., 2001).

O *Bubble Rap* (ALMEIDA, 2011 apud HUI et al., 2008) utiliza a métrica de centralidade no grafo da rede para encontrar os nós que interligam as comunidades. As comunidades de nós são identificadas por um algoritmo. Uma vez que a mensagem se encontra na comunidade correta as informações são propagadas utilizando uma métrica de centralidade para encontrar o melhor nó para repassar a mensagem.

Foi apresentado em (ALMEIDA, 2011 apud MOTA et al., 2009) um algoritmo de roteamento para redes de emergência. Neste trabalho foi considerado que os nós se movimentam em grupos e o repasse de agregados é realizado de forma hierárquica. É selecionado um líder para o grupo de nós e este líder é responsável por repassar as mensagens para os demais grupos.

O mais simples protocolo de roteamento para DTNs é o *Direct Transmission* (ALMEIDA, 2011 apud SPYROPOULOS et al., 2005a), neste é mantido apenas uma cópia do agregado na rede e um nó encaminha a mensagem apenas se encontrar o destinatário final. Este algoritmo depende da mobilidade dos nós para que o nó origem entre em contato com o nó destino. Este protocolo apresenta o maior atraso dentre os protocolos DTN e como vantagem apresenta o menor consumo de energia.

2.7 Spray and Wait

Quando (SPYROPOULOS et al. 2005) propôs este protocolo de roteamento, procurava-se atingir certos objetivos desejáveis como: executar um número significativamente menor de transmissões do que o epidêmico e outros esquemas de roteamento baseado em inundações, sob todas as condições; gerar baixa contenção, especialmente sob cargas de alto tráfego; obter uma taxa de atraso melhor do que aos dos esquemas já existentes; ser altamente escalável, ou seja, manter o desempenho acima apesar das mudanças no tamanho da rede ou densidade de nós; e por fim, ser simples e requerer o mínimo possível de conhecimento sobre a rede, a fim de facilitar a sua aplicação.

Desse modo, o *Spray and Wait* busca combinar a velocidade do roteamento epidêmico com a simplicidade e economia do *Direct Transmission*. Ele reduz a sobrecarga da rede, pois reduz as transmissões de agregados, porém é capaz de manter uma alta taxa de

entrega bem como uma baixa latência. Segundo (SPYROPOULOS et al. 2005) o roteamento do *Spray and Wait* consiste em duas fases:

- Fase *Spray*: para cada mensagem originária de um nó fonte, L cópias da mensagem são espalhadas ou encaminhadas para os nós vizinhos.
- Fase *Wait*: se o destino não foi encontrado na primeira fase, cada um dos L nós portadores da mensagem fazem a entrega utilizando o algoritmo *Direct Transmission* (ou seja, irá encaminhar a mensagem apenas para o seu destino).

Inicialmente o "salto" é iniciado ao se espalhar cópias da mensagem de forma semelhante ao roteamento epidêmico. Quando cópias suficientes forem espalhadas para garantir que pelo menos um deles vá encontrar o destino de forma rápida (com alta probabilidade), o nó fonte da mensagem para e permite que cada nó carregando uma cópia realize a transmissão direta (SPYROPOULOS et al. 2005).

Em (SPYROPOULOS et al. 2005), foi demonstrado que o algoritmo limita o número de cópias e transmissões por mensagem sem comprometer a performance. Usando teoria e simulação foi observado que: sob o cenário de baixa carga, o *Spray and Wait* resulta em muito menos transmissões e atrasos quando comparado com esquemas baseados em inundações; sob alta carga, ele produz significativamente melhores atrasos e menos transmissões do que esquemas baseados em inundações; é altamente escalável, exibindo bom e previsível desempenho para uma grande variedade de tamanhos de rede, densidades de nós e níveis de conectividade; e a medida que o tamanho da rede e o número de nós de aumenta, o número de transmissões por nó que o *Spray and Wait* requer a fim de conseguir o mesmo desempenho diminui. Além disso, foi demonstrado que ele pode ser facilmente ajustado para atingir determinados requisitos de QoS, mesmo em redes desconhecidas.

Para se utilizar este esquema não é necessário que os nós possuam conhecimento da rede. Este protocolo apresenta seus melhores resultados quando a rede apresenta alta mobilidade, ou seja, se a mobilidade for pequena e localizada afetará o desempenho negativamente (ALMEIDA, 2011).

3 Análise Quantitativa de Desempenho do Spray and Wait

Neste capítulo será apresentada em detalhes como foi efetuada a análise quantitativa de desempenho do *Spray and Wait*.

3.1 Ambiente de Simulação

A avaliação do desempenho do algoritmo de roteamento escolhido neste trabalho foi realizada através do uso do simulador *The Opportunistic Network Environment* (The ONE) (KERANEN et al, 2009) na versão 1.5.1-RC2, desenvolvido pelo *Networking Laboratory* da Universidade de Tecnologia de Helsinki, para estudo de mobilidade em DTNs.

O The ONE é um simulador de ambientes capaz de gerar a movimentação dos nós usando diferente modelos de mobilidade; rotear mensagens entre nós com vários algoritmos de roteamento para DTNs e diferentes tipos de emissores e receptores; visualizar tanto as transmissões das mensagens como a movimentação dos nós em tempo real por meio de uma interface de usuário; pode importar dados de mobilidade a partir de registros do mundo real ou de outros geradores de mobilidade; pode produzir uma variedade de relatórios de movimentação dos nós, geração e transmissão de mensagens, bem como estatísticas gerais. O simulador foi escrito em Java 1.6 e tem sua documentação disponível no site do projeto dele. (TKK/COMNET, 2015)

Desse modo, observa-se que o The ONE é uma ferramenta poderosa para simular diversas situações específicas de movimentações de nós em caminhos pré-determinados ou aleatórios, sendo possível customizar a execução conforme seja necessário.

3.1.1 Funcionamento e Características

Antes de rodar o simulador e executar as simulações, faz-se necessários seguir um conjunto de procedimentos fornecidos pela documentação do próprio que pode variar conforme o sistema operacional ou a necessidade do usuário. Por exemplo, se o usuário desejar, poderá executar o arquivo “one.bat” diretamente de um *prompt* de comando ou poderá executá-lo a partir de um compilador de sua preferência. Além disso, é possível

executar as simulações sem a necessidade da interface, a documentação explica como realizar este procedimento.

O simulador já vem com uma configuração pré-definida de modelos de mobilidade, grupos de nós, entre outros parâmetros, e pode ser executado sem necessidade de alterações adicionais.

Um cenário pode ser construído por meio da inserção de diversos elementos, dentre eles: grupos de nós, eventos, protocolos de roteamento, modelos de movimentação, relatórios e mapas.

Cada grupo de nós possui características próprias, tais como a velocidade de deslocamento, a taxa de transmissão, o tamanho de *buffer*, a quantidade de nós e um identificador. Diversos grupos podem ser inseridos no cenário, possibilitando uma variedade de nós presentes no ambiente de simulação. Os eventos a serem gerados na simulação nada mais são do que mensagens (ou pacotes) com origem e destino determinados. Eles podem ser configuráveis em tamanho e intervalo de tempo de criação, além de poderem restringir quais nós podem gerá-los e recebê-los. Os protocolos de roteamento podem ser aplicados de maneira global ou local no ambiente de simulação, ou seja, cada grupo distinto de nós pode possuir seu próprio protocolo de roteamento. No ONE, por padrão, já estão implementados os protocolos *Epidemic*, *Spray And Wait*, *MaxProp* e *PRoPHET* (TABARELLI et al., 2009).

Os modelos de mobilidade são padrões de deslocamento dos nós e também podem ser aplicados de forma global ou local aos grupos de nós. Dentre os diversos modelos disponíveis, encontram-se: o *CarMovement* que é um modelo para carros, no qual o nó possui um comportamento de seguir para um destino a partir de uma origem; o *BusMovement* que é um modelo para ônibus, no qual o nó possui um comportamento que alterna origem e destino continuamente (comportamento ping-pong); o *ShortestPath*, modelo em que o nó possui dois pontos de origem e destino, e a menor rota entre tais pontos é calculada usando o algoritmo de Dijkstra, e o *RandomWaypoint*, modelo em que são criados diversos caminhos aleatórios entre pares de pontos, com o nó se deslocando por estes caminhos (TABARELLI et al., 2009).

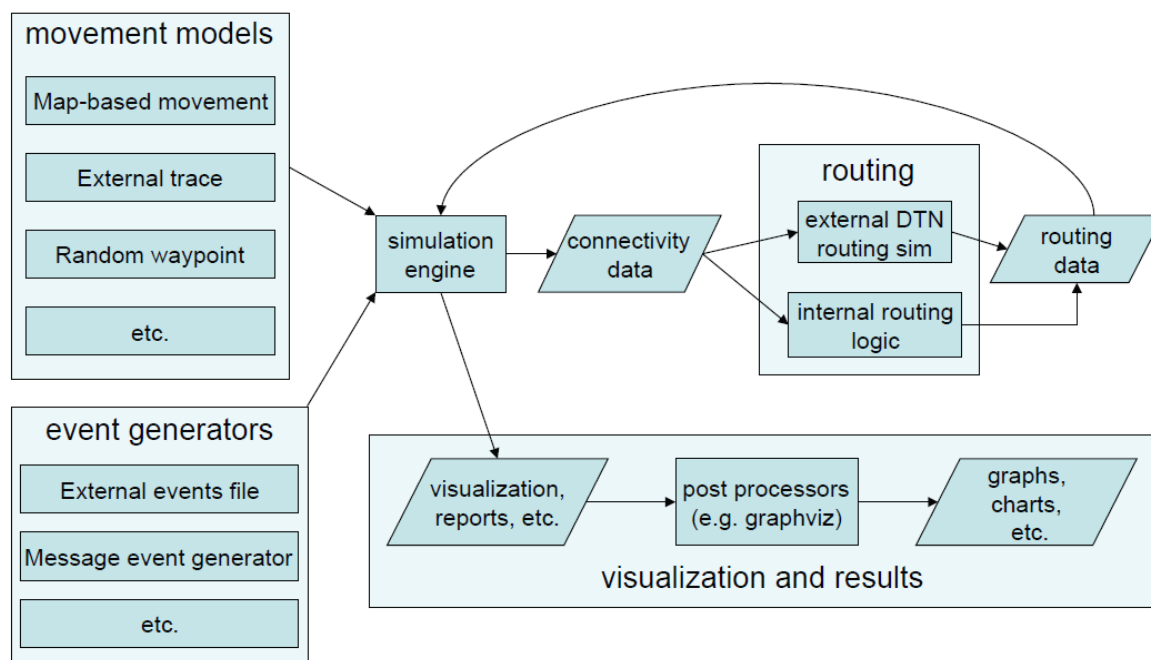


Figura 5 - Arquitetura do simulador The ONE (KERANEN et al, 2009)

Os relatórios são gerados a partir de cada rodada de simulações. Eles fornecem estatísticas essenciais para a interpretação e análise dos diferentes cenários construídos. Por meio deles é possível obter o número de mensagens criadas durante o intervalo de simulação; o número de mensagens entregues; bem como o número de mensagens que alcançaram com sucesso seu destinatário; e dados sobre a conectividade da rede, como instantes de ocorrência de conexão e desconexão entre nós.

Os mapas representam o conjunto de pontos e retas nos quais os nós podem se deslocar. No simulador ONE, é utilizado um formato específico de representação de mapas, o formato WKT. Baseando-se no fluxo dos dados, a figura 5 apresenta a estrutura de funcionamento do ONE.

3.1.2 Interface

A interface do ONE é composta por quatro áreas principais: área da simulação, controle da simulação, lista dos nós e log de eventos. A figura 6 apresenta a interface gráfica e as quatro áreas:

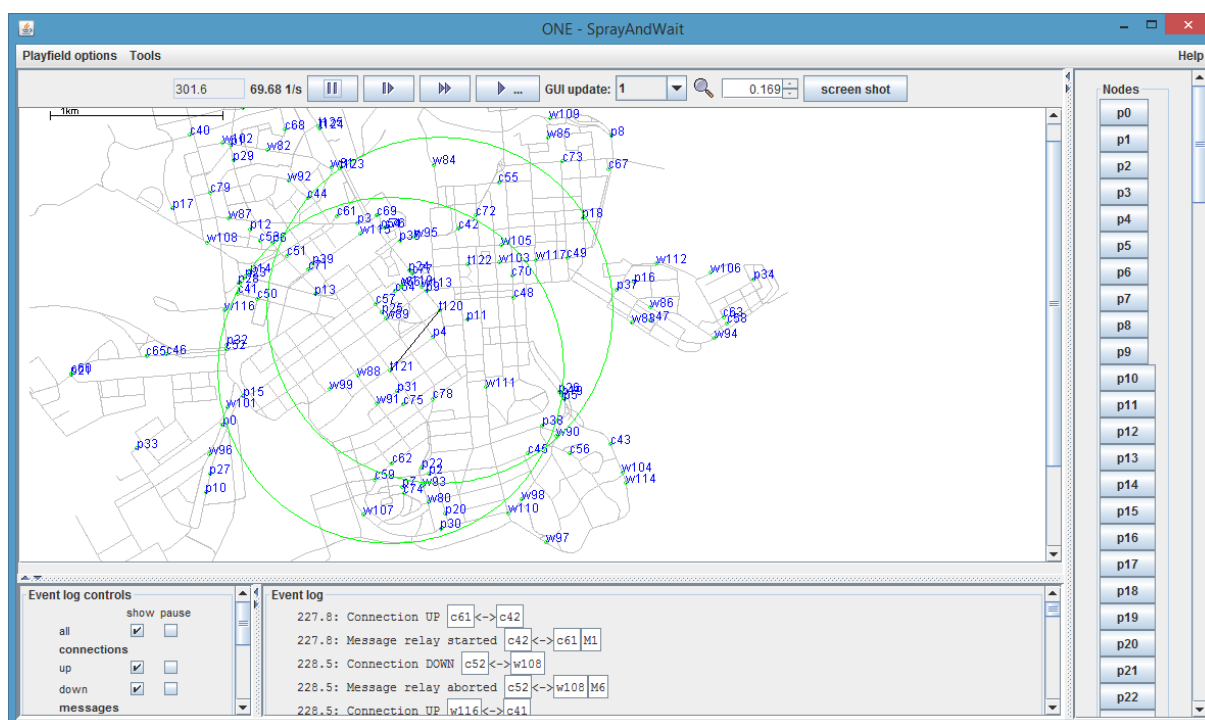


Figura 6 - Interface gráfica do simulador The ONE

Na parte superior da interface, observa-se os botões para iniciar, acelerar, pausar a simulação, efetuar um *screenshot* da tela, ajustar a velocidade de atualização da interface gráfica e o nível de zoom da área de simulação.

Na parte inferior da interface, encontra-se o log de eventos que apresenta as informações sobre conexões e mensagens, como o instante de início da conexão, o processo de custódia de uma mensagem e o término da conexão entre dois nós. É possível filtrar as mensagens do log e ainda acessar informações mais detalhadas sobre algum nó diretamente pela área de eventos.

Na área principal, são exibidos os nós, os caminhos em que podem trafegar e o raio de transmissão de cada um deles; é possível notar que, durante a custódia de uma mensagem, os dois nós envolvidos permanecem ligados por um segmento de reta, indicando tal evento.

À direita, situa-se a lista dos nós do cenário; é possível acessar informações mais detalhadas sobre cada um deles clicando em sua identificação; com isso, visualizam-se quantas mensagens estão no *buffer*, com quais nós está conectado e o status das suas mensagens geradas ou em custódia. Além disso, a figura 7 mostra a interface gráfica com a rota atual do nó em vermelho.

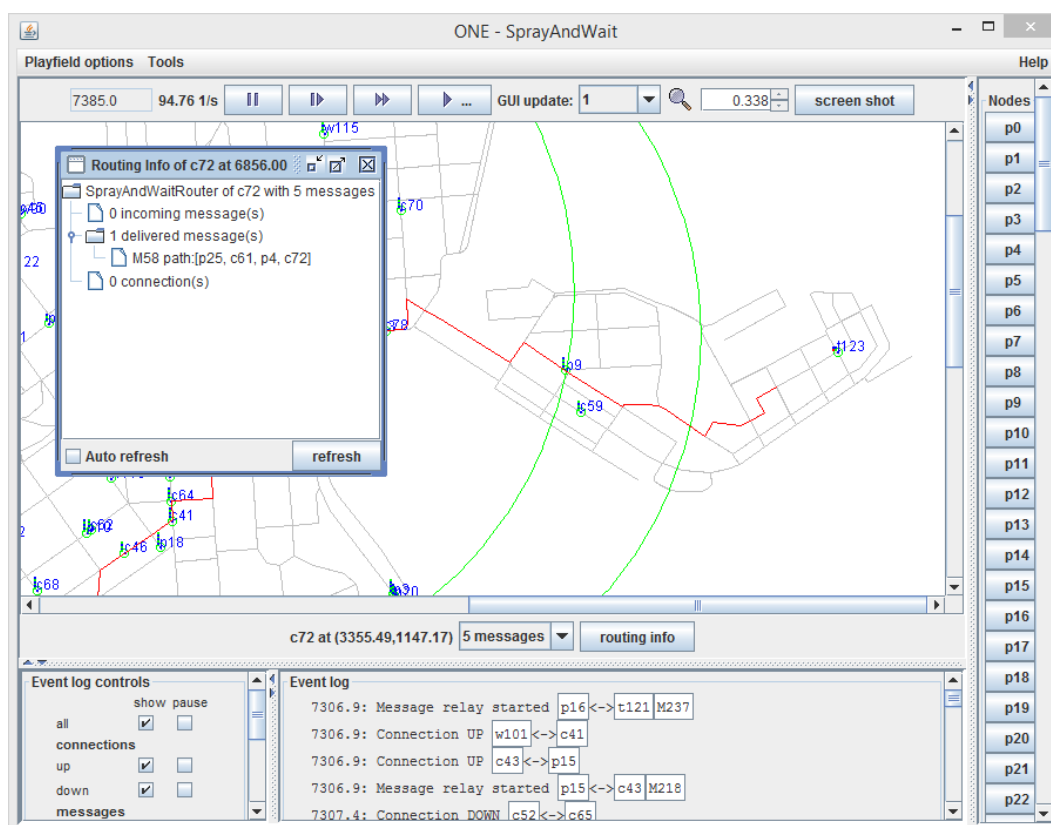


Figura 7 - Detalhes de um nó específico no simulador ONE e sua a rota de movimentação.

3.2 Cenários e Parâmetros das Simulações

Para avaliar o desempenho do algoritmo *Spray and Wait*, algumas métricas de desempenho serão definidas nesta seção e utilizadas nas seções seguintes para apresentar e discutir a eficiência do algoritmo diante dos cenários propostos.

Probabilidade de entrega das mensagens ao destino. Essa métrica é definida como sendo a razão entre as mensagens que foram roteadas até o nó destino e o total de mensagens criadas na rede (FERNANDES, 2009).

O **atraso médio** é a soma dos atrasos de todas as mensagens trafegadas na rede sobre a quantidade dessas mensagens. Onde o atraso é definido como a diferença entre o instante de tempo que a mensagem foi entregue ao destino e o instante de tempo de criação da mesma. Esta métrica é importante, dado que se o roteamento da origem ao destino for muito longo, a mensagem pode se tornar obsoleta (FERNANDES, 2009).

Há ainda outras métricas a ressaltar que são: a quantidade de mensagens entregues durante a simulação; a porcentagem de mensagens descartadas por falta de rota interligando

origem e destino; e o número de contatos que representam as transmissões entre os nós. (GOLASH, 2015)

3.2.1 Cenários

Com objetivo de avaliar o desempenho do algoritmo *Spray and Wait* em DTNs com características de redes móveis foram simulados três cenários: urbano, veicular e o cenário de desastres. A cidade de Helsinki foi escolhida para a simulação, pois a maior parte das configurações necessárias para definir as características desta cidade já vem definidas no cenário padrão do simulador The ONE. A figura 8 mostra o mapa que já vem definido no projeto do simulador.

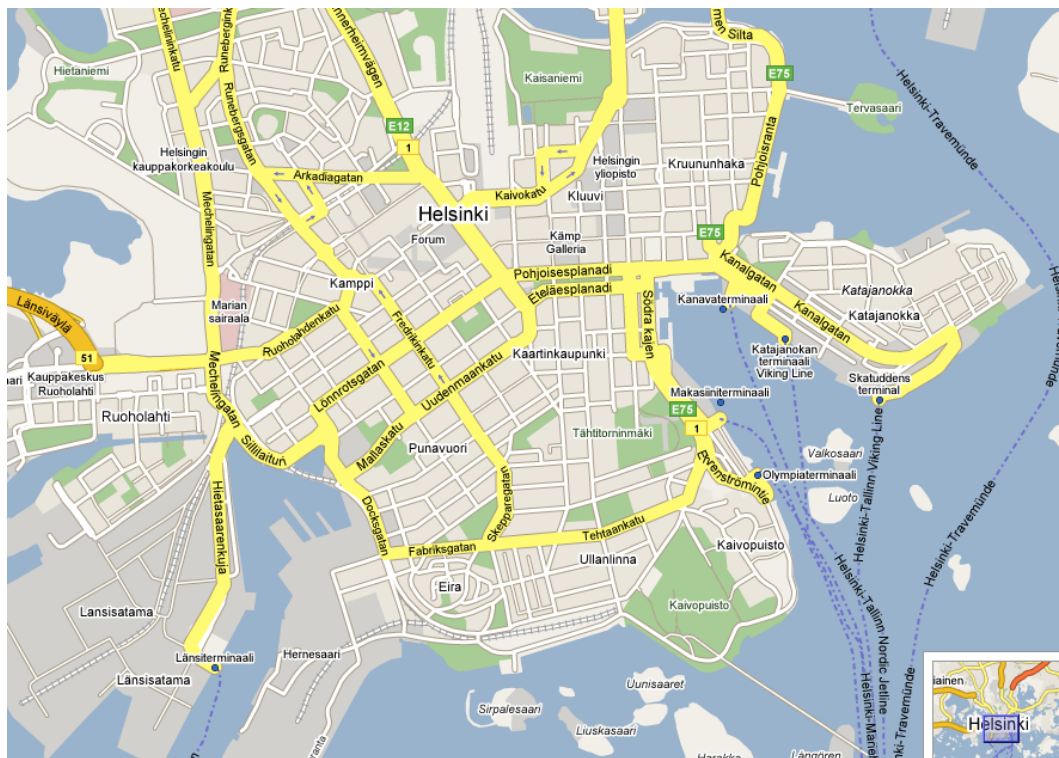


Figura 8 - Mapa da cidade de Helsinki, Finlândia.

3.2.1.1 Cenário Urbano

No caso do cenário urbano, buscou-se fazer uma analogia de um cenário real, ao qual houvesse pedestres e carros que se movimentassem ao longo da cidade supracitada, tanto em ruas como em avenidas, shoppings, escolas, entre outros; e que cada um deles tivesse a

capacidade de se comunicar ou transmitir dados por meio de dispositivos que tivessem a tecnologia Wi-Fi Direct embarcada neles.

Para simular o cenário urbano, 300 nós foram definidos e divididos em dois grupos, o primeiro foi composto por pedestres e o segundo grupo por carros. Estes grupos foram colocados em uma área de 4500 metros de largura e 3400 metros de comprimento, essas medidas se referem à área central da cidade de Helsinki, na Finlândia (NUNES, 2010).

Entre os modelos de mobilidade baseados em mapa disponíveis no simulador The ONE, nota-se que o *Shortest Path Map-based Movement* (SPMBM) (MOSCHETTO et al., 2011 apud KERÄNEN et al., 2009] tem um comportamento semelhante ao *Levy Walk* no que diz respeito à seleção de menor caminho, utilizando o algoritmo de Dijkstra. Portanto, a escolha do SPMBM pode ser considerada razoável devido ao comportamento dos seres humanos que na maioria das vezes procuram escolher o menor caminho para se chegar a um destino e ao conhecimento prévio que eles possuem em relação à cidade onde vivem.

Com relação às velocidades de deslocamento dos nós, foram definidas de 0,4 m/s a 1,0 m/s para as pessoas e de 2,4 m/s a 10 m/s para os automóveis. Também foram definidas duas interfaces de comunicação, uma para cada um dos grupos. Cada interface possui ainda mais duas configurações: o raio (ou alcance) e a velocidade de transmissão.

O raio de alcance dos nós foi definido em 30 metros para as pessoas e em 80 metros para os automóveis com o intuito de representar o alcance da tecnologia Wi-Fi Direct, uma vez que os pedestres poderiam andar em diferentes ambientes com muitos obstáculos, como por exemplo, postes, paredes, monumentos entre outros tipos de materiais que possam interferir na transmissão dos dados; já os automóveis podem transitar em rodovias, pistas e avenidas, ou seja, movimentam-se em ambientes abertos ou com menos obstáculos e interferências. A velocidade de transmissão das interfaces foi definida em aproximadamente 3 Mbps simulados.

Além destas configurações, foi definido que cada simulação durasse 55 horas simuladas. Este valor foi definido através de um estudo para tornar o resultado da simulação o mais exato possível (NUNES, 2010). Como o simulador é configurado em segundos, as 55 horas são equivalentes a 198.000 segundos simulados. Outro parâmetro definido foi o intervalo de tempo de criação de mensagens com o valor igual a 40 segundos simulados, isso significa que no final da simulação foram geradas 4950 mensagens.

Por fim, para uma maior avaliação deste trabalho, o número de cópias da mensagem na rede foi variado. Inicialmente 2 cópias foram definidas, depois 6 cópias e por fim, 10

cópias da mensagem, consideremos isso como uma rodada de simulações. Além disso, numa segunda série de simulações deste mesmo cenário, foi realizada a mudança do parâmetro de variação do tempo em que os nós devem permanecer no mesmo lugar após chegar ao local de destino e antes de partir para um novo local. Desse modo, o intervalo de tempo que no primeiro caso foi de 0 a 120 segundos passou a ser de 0 a 14400 segundos, o que representa 4 horas simuladas. Esse parâmetro foi alterado visando aproximar a simulação ainda mais à realidade, pois muitas pessoas e carros não tem uma mobilidade tão alta como representada na primeira rodada de simulações. A tabela 1 apresenta os parâmetros gerais utilizados na simulação do cenário urbano e o apêndice A contém o script de configuração da simulação.

Parâmetro	1 ^a Rodada	2 ^a Rodada
Tempo de Simulação	198.000s	198.000s
Interfaces de Comunicação	2	2
Velocidade de Transmissão	3Mbps	3Mbps
Raio de Transmissão	[35,95] m	[35,95] m
Número de Grupos	2	4
Número de Nós	300	300
Modelo de Mobilidade dos Nós	SPMBM	SPMBM
Tamanho do Buffer dos Nós	5MB	5MB
Tempo de Espera (<i>waitTime</i>)	Entre 0s e 120s	Entre 0s e 14400s
Geradores de Eventos	1	1
Classe do Gerador de Eventos	MessageEventGenerator	MessageEventGenerator
Intervalo de Criação	40s	40s
Tamanho das Mensagens	Entre 500k e 1MB	Entre 500k e 1MB
Nós geradores de mensagens	Todos	Todos
Área de Simulação	4500m x 3400m	4500m x 3400m
Número de Cópias da Mensagem	[2, 6, 10]	[2, 6, 10]
Quantidade de Simulações	4	4
Algoritmo de Roteamento	SprayAndWait	SprayAndWait

Tabela 1 - Parâmetros gerais da simulação para o cenário urbano.

3.2.1.2 Cenário Veicular

No caso do cenário veicular, buscou-se fazer uma analogia de um cenário real, ao qual houvesse diversos tipos de veículos que se movimentassem ao longo da cidade supracitada, tanto em ruas como em avenidas; e que cada um deles tivesse a capacidade de se comunicar ou transmitir dados por meio de dispositivos que tivessem a tecnologia Wi-Fi Direct embarcada neles.

A maioria dos parâmetros definidos para simular o cenário veicular são os mesmos definidos para o cenário urbano, sendo estes: a quantidade de nós, a área ou cidade simulada, o modelo de mobilidade, a duração da simulação, o intervalo de criação das mensagens e a velocidade de transmissão dos dados. Além destes, também foi feita uma variação no número de cópias da mensagem na rede.

Todavia, neste caso, os 300 nós foram divididos em 7 grupos: o primeiro foi composto por 100 nós, estes representando os carros; o segundo foi composto por 50 nós, estes representando os caminhões; o terceiro foi composto por 44 nós, estes representando os ônibus; o quarto foi composto por 100 nós, estes representando as motos; o quinto, o sexto e o sétimo foram compostos por 2 nós cada um, estes representando os bondinhos elétricos. Ressaltando-se que os grupos dos bondinhos poderiam representar outros tipos de veículos de transporte em massa, como metrô, trens, BRTs (*Bus Rapid Transit*), entre outros.

Com relação aos grupos que representam meios transporte público, que neste caso são: o grupo de ônibus e os grupos dos bondinhos, uma outra configuração foi definida, sendo esta um arquivo com uma rota específica para cada grupo. Já com relação a interface, apenas uma foi definida e seu raio de alcance permaneceu o mesmo, 80 metros.

Quanto à velocidade de deslocamento dos veículos, foram definidas velocidades mínimas e máximas para cada grupo, sendo estas: de 2,4m/s a 20m/s para os carros; de 2,4m/s a 14m/s para os caminhões, de 2,4m/s a 16m/s para os ônibus; de 2,4m/s a 20m/s para as motos; e de 2,4m/s a 20m/s para os bondinhos. Para o parâmetro que contém o intervalo de tempo em que os nós devem permanecer no mesmo local antes de partir para um novo destino, foi definido que os meios de transporte ficariam de 10 a 30 segundos e os outros grupos ficariam de 0 a 120 segundos. A tabela 2 apresenta os parâmetros gerais utilizados na simulação do cenário veicular e o apêndice B contém o script de configuração da simulação.

Parâmetro	Simulação
Tempo de Simulação	198.000s
Interfaces de Comunicação	2
Velocidade de Transmissão	3Mbps
Raio de Transmissão	[35,95] m
Número de Grupos	7
Número de Nós	300
Modelo de Mobilidade dos Nós	SPMBM
Tamanho do <i>Buffer</i> dos Nós	5MB
Tempo de Espera (<i>waitTime</i>)	Entre 0s e 120s
Geradores de Eventos	1
Classe do Gerador de Eventos	MessageEventGenerator
Intervalo de Criação	40s
Tamanho das Mensagens	Entre 500k e 1MB
Nós geradores de mensagens	Todos
Área de Simulação	4500m x 3400m
Número de Cópias da Mensagem	[2, 6, 10]
Quantidade de Simulações	4
Algoritmo de Roteamento	SprayAndWait

Tabela 2 - Parâmetros gerais da simulação para o cenário veicular.

3.2.1.3 Cenário de Emergência

No caso do cenário de emergência (ou desastre), buscou-se fazer uma analogia de um cenário real, ao qual houvesse equipes de emergência de diversas categorias executando atividades de socorro às vítimas de algum desastre, seja ele um furacão, terremoto ou qualquer outro incidente que fizesse cessar o fluxo normal do cotidiano de uma cidade.

Na simulação, supõe-se que o desastre aconteceu na mesma cidade citada anteriormente. Diferente dos outros cenários, os quais não tinham restrições de movimentação, neste caso as equipes de busca e resgate se movimentam entre pontos de interesse (do inglês, *Points Of Interest* - POI) da cidade, como regiões de busca e resgate. Para este cenário foram definidos os seguintes pontos de interesse: um centro de comando e controle; dois hospitais; e locais com muitas pessoas que necessitam de ajuda, como estações de transbordo, escolas ou shoppings. Todavia, as equipes podem se deslocar para outros

pontos da cidade que sejam de menos interesse, mas a probabilidade de se deslocarem para um ponto diferente dos já especificados é muito pequena.

Assim como nos outros cenários, cada um dos membros das equipes tem a capacidade de se comunicar ou transmitir dados por meio de dispositivos que contém a tecnologia Wi-Fi Direct embarcada neles. Uma diferença a ser ressaltada para este cenário é que o centro de controle e os dois hospitais também tem a capacidade de se comunicar ou transmitir dados com os membros das equipes. Outra diferença é que para este cenário, buscou-se mesclar a alta mobilidade dos veículos e com a baixa mobilidade dos pedestres, ou seja, no caso dos “pedestres” das equipes houve uma divisão entre aqueles que poderiam estar se deslocando entre os pontos de interesse e aqueles que poderiam estar executando alguma atividade que demandasse maior tempo num local a fim de fazer uma busca ou executar um resgate.

Quanto aos demais parâmetros da simulação, a maioria deles foi definida como no cenário urbano. As diferenças se deram pela alteração na distribuição dos nós em outros grupos específicos. Assim, os 300 nós ficaram divididos nos seguintes grupos: três grupos estacionários contendo apenas um nó, que representam, cada um, o centro de comando e controle e os dois hospitais; um grupo de carros com 30 nós; um grupo de caminhões contendo 10 nós; um grupo de pedestres com alta mobilidade contendo 120 nós; um grupo de pedestres com baixa mobilidade contendo 107 nós; e um grupo de motos contendo 30 nós. Como os três grupos estacionários não se movimentam, foi preciso alterar o modelo de mobilidade para o *StationaryMovement*.

Foi preciso utilizar o software OpenJUMP, na versão 1.8.0, para definir os pontos de interesse no mapa da cidade de Helsink. O OpenJUMP é um Sistema de Informação Geográfica (SIG) codificado em Java, desenvolvido e mantido por um grupo de voluntários e é baseado no JUMP GIS desenvolvido pela empresa Vivid Solutions (JUMP, 2015). A figura 9 apresenta a interface da ferramenta OpenJUMP com o mapa de rotas e os pontos de interesse.

Por fim, além de definir os pontos de interesse no mapa, foi necessário definir a probabilidade para os grupos de nós escolherem os pontos de interesse como destino. Para o grupo de carros, caminhões, pedestres e motos foram definidas as mesmas probabilidades, sendo estas: 40% para o centro de controle e hospitais e 20% para os demais pontos de interesse onde há maior necessidade dos trabalhos das equipes.

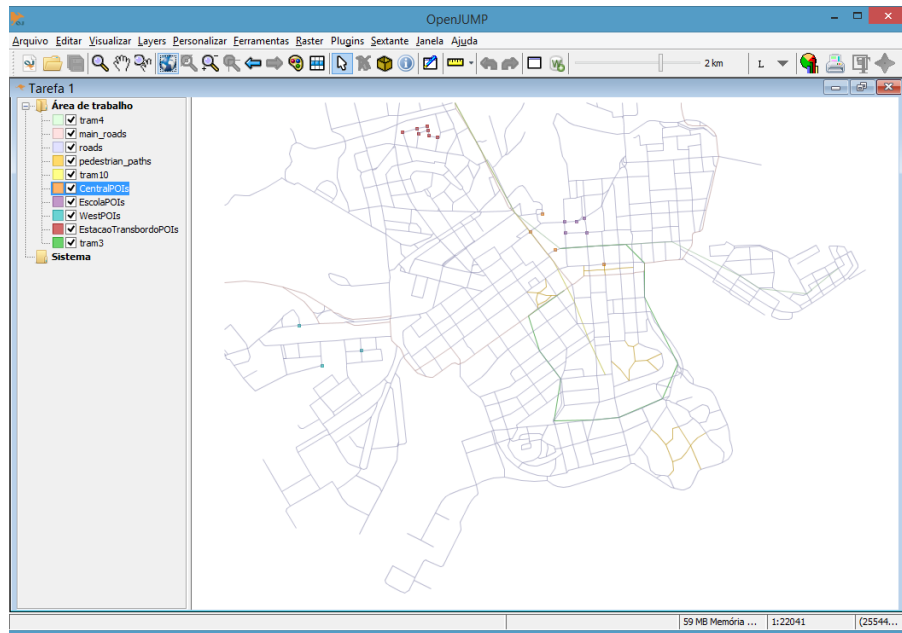


Figura 9 - Interface da ferramenta OpenJUMP com as rotas e pontos de interesse da cidade de Helsink.

A tabela 3 apresenta os parâmetros gerais utilizados na simulação do cenário de emergência e o apêndice E contém o script de configuração do algoritmo *Spray and Wait*.

Parâmetro	Simulação
Tempo de Simulação	198.000s
Interfaces de Comunicação	2
Velocidade de Transmissão	3Mbps
Raio de Transmissão	[35,95] m
Número de Grupos	8
Número de Nós	300
Modelo de Mobilidade dos Nós	SPMBM e StationaryMovement
Tamanho do Buffer dos Nós	5MB
Tempo de Espera (waitTime)	Entre 0s e 120s
Geradores de Eventos	1
Classe do Gerador de Eventos	MessageEventGenerator
Intervalo de Criação	40s
Tamanho das Mensagens	Entre 500k e 1MB
Nós geradores de mensagens	Todos
Área de Simulação	4500m x 3400m
Número de Cópias da Mensagem	[2, 6, 10]

Quantidade de Simulações	4
Algoritmo de Roteamento	SprayAndWait

Tabela 3 - Parâmetros gerais da simulação para o cenário de emergência.

3.3 Avaliação dos Resultados

Após a definição dos parâmetros e configuração do simulador para cada cenário, as simulações foram executadas. É importante ressaltar que cada simulação foi executada quatro vezes. No entanto, como a probabilidade se manteve a mesma, foi utilizado apenas um relatório gerado por cada simulação para construir os gráficos que serão apresentados a seguir.

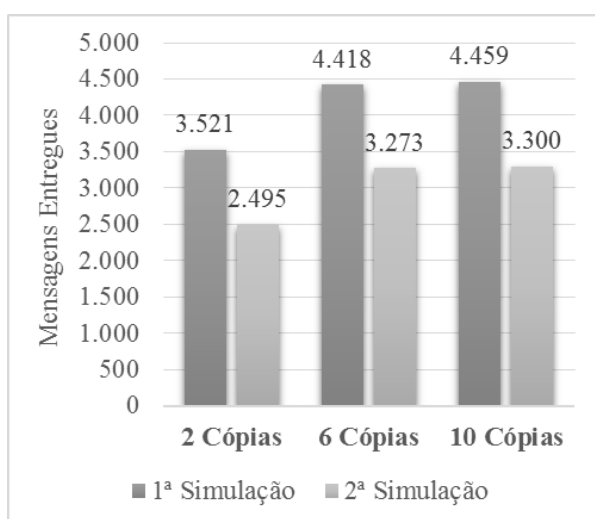


Gráfico 1 - Quantidade de mensagens entregues por simulação.

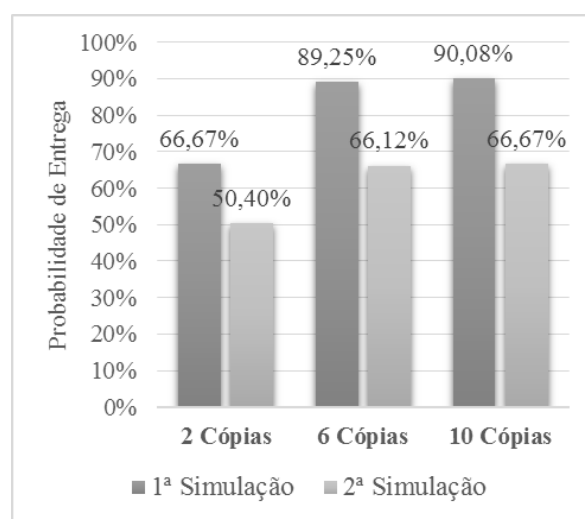


Gráfico 2 - Probabilidade de entrega da mensagem por simulação.

Os gráficos de 1 a 5 foram construídos com base nos relatórios das simulações do cenário urbano. Como expresso anteriormente, foram realizadas duas rodadas de simulações. Cada rodada de simulações equivale as simulações com variações no número de cópias de mensagens, sendo estas: 2, 6 e 10 cópias. A diferença entre as duas rodadas de simulações se deve a alteração no parâmetro que define o tempo em que um nó deve permanecer no local de destino após alcançá-lo.

Como se pode observar nos gráficos de 1 a 5, o aumento do intervalo no parâmetro de tempo de espera dos nós resultou em queda dos valores em cada um dos resultados. Desse

modo, apenas com estes primeiros 5 gráficos, já ficou claro que para o *Spray and Wait* obter bons resultados é preciso que haja alta e constante mobilidade dos nós.

No caso dos gráficos 1 e 2 os valores caíram significativamente, verificando-se que houve uma redução média de 15% na entrega das mensagens geradas e na probabilidade de entrega. Percebe-se também que quanto maior o número de cópias da mensagem na rede, maior será a quantidade de mensagens entregues. Apesar disso, mesmo com um número baixo de cópias, no pior caso a probabilidade de entrega ainda ficou acima dos 50%, em virtude de que na simulação com apenas duas cópias e intervalo de tempo de espera elevado, das 4950 mensagens geradas, 2495 mensagens ainda foram entregues.

Ainda analisando os dois primeiros gráficos, percebe-se que embora as simulações com 10 cópias tenha obtido os melhores resultados, os seus valores não ficaram distantes dos valores das simulações com 6 cópias. A diferença entre os valores para a probabilidade de entrega da mensagem foi menor do que 1%.

Embora o cenário urbano esteja sendo analisado neste momento, é importante salientar que este dado, a pequena diferença nos resultados, é importante principalmente para sistemas que procuram gastar o mínimo de energia dos nós ou possuem uma base de armazenamento pequena, como por exemplo um sistema de monitoramento de qualquer tipo de região ou evento, pois um número menor de cópias na rede resulta em menos transmissões e menos armazenamento.

Outra informação observada com relação a quantidade de cópias é que ao aumentar de 2 para 4 a quantidade de cópias da mensagem, a quantidade de mensagens entregues e a probabilidade de entrega são elevados em quase 13%, contudo ao aumentar mais 4 cópias, os valores não mantêm o mesmo ritmo anterior, resultando desse modo numa estabilização do desempenho do algoritmo.

Analisando os gráficos 3 e 4, observa-se respectivamente a quantidade de transmissões bem sucedidas e a quantidade de mensagens apagadas por simulação. Verifica-se que o crescimento nos dois casos são proporcionais, tanto na primeira como na segunda rodada de simulações. No caso do gráfico 3, percebe-se que a quantidade de transmissões aumentou numa média de 20.000 a cada momento que se elevou o número de cópias. O mesmo comportamento é perceptível no gráfico 4, onde o número de mensagens apagadas aumenta numa média de 20.000 a cada vez que se eleva o número de cópias. Ressaltando-se que neste caso, as mensagens são apagadas por que o *buffer* de um nó está cheio, desse modo, quando uma nova mensagem é recebida, a mais antiga é descartada.

Como observado anteriormente, verificou-se que 6 cópias da mensagem da rede já seria um valor equilibrado para se escolher como parâmetro, pois o desempenho do algoritmo tende a se manter estável. Desta maneira, com base nos gráficos 3 e 4, a escolha de 10 cópias da mensagem na rede resultaria em quase o dobro transmissões desnecessárias, gerando assim, desperdício de energia dos nós. E o mesmo acontece com relação número de mensagens apagadas quando se escolhe 10 cópias de mensagens, resulta em quase o dobro de mensagens apagadas.

Observando o gráfico 5, verifica-se a latência média em segundos por rodada de simulações. Este gráfico mostra que a latência vai caindo a medida que o número de cópias da mensagem vai subindo. Ressaltando-se que essa latência representa o tempo entre a criação da mensagem até a entrega dela. Desse modo, no caso mais otimista, que é a simulação com alta mobilidade dos nós e a definição de 10 cópias da mensagem, a latência média foi de 2377 segundos, o que equivale a mais de 39 minutos de atraso até que a entrega seja efetuada.

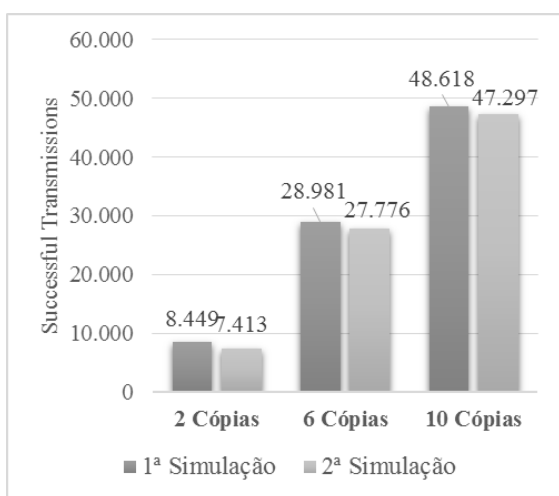


Gráfico 3 - Quantidade de transmissões bem sucedidas por simulação.

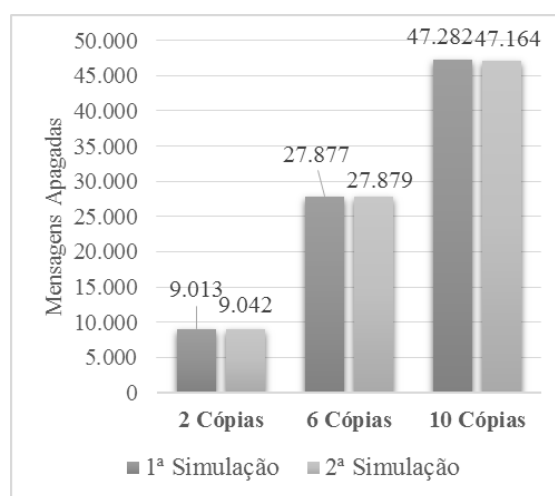


Gráfico 4 - Quantidade de mensagens apagadas por simulação.

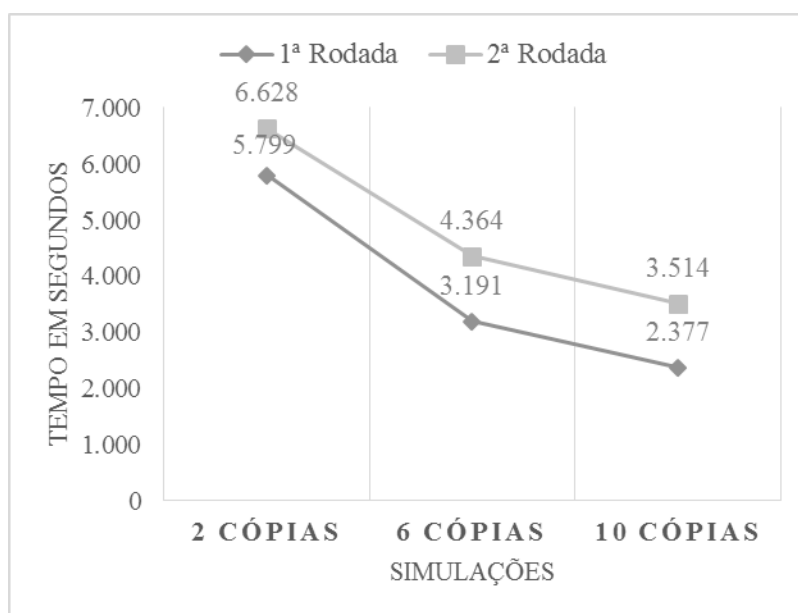


Gráfico 5 - Latência Média em segundos por rodada de simulações.

Os gráficos de 6 a 10 foram construídos com base nos relatórios das simulações do cenário veicular. Diferentemente do que foi feito no cenário urbano, neste cenário buscou-se analisar apenas o desempenho do algoritmo levando em conta a alta mobilidade dos nós.

Nos gráficos 6 e 7 são apresentados respectivamente a quantidade de mensagens e entregues e a probabilidade de entrega de uma mensagem. Analisando-os, percebe-se que, independentemente do número de cópias das mensagens na rede, o desempenho do algoritmo foi ótimo. Nas três simulações a probabilidade de entrega da mensagem ficou acima dos 99%, ou seja, mesmo no pior caso, que foi a simulação com duas cópias, das 4950 mensagens criadas, apenas 35 mensagens não foram entregues.

Nos gráficos 8 e 9, são apresentados respectivamente a quantidade de transmissões bem sucedidas por simulação e a quantidade de mensagens apagadas por simulação. Observa-se que os resultados cresceram proporcionalmente, ou seja, mantiveram um padrão muito semelhante aos resultados do cenário urbano.

Até este momento, a escolha de 2 cópias da mensagem na rede parece ser a mais coerente e equilibrada. No entanto, ao se observar o gráfico 10 é perceptível a diferença da latência média nas duas primeiras simulações. Da simulação com apenas 2 cópias para a simulação com 6 cópias, há uma redução em mais de 50%, ou seja, no primeiro caso a latência foi de 1546 segundos, o que equivale a mais de 25 minutos, e no segundo caso a latência foi de 602 segundos, o que equivale a um pouco mais de 10 minutos. Na terceira

simulação, com 10 cópias, a redução já é menor, passando para 431 segundos, o que equivale a um pouco mais que 7 minutos.

Desta maneira, dependendo dos interesses envolvidos ao se construir um sistema de comunicação ou monitoramento, ou o número de cópias por de ser mais importante quando se pretende economizar os recursos dos nós, ou a latência na entrega da mensagem pode ser mais importante quando se procura maior rapidez na comunicação. Ressaltando-se que estes resultados são baseados num cenário de alta mobilidade. Além disso, este cenário é o que tem a mobilidade mais alta entre os três citados neste trabalho.

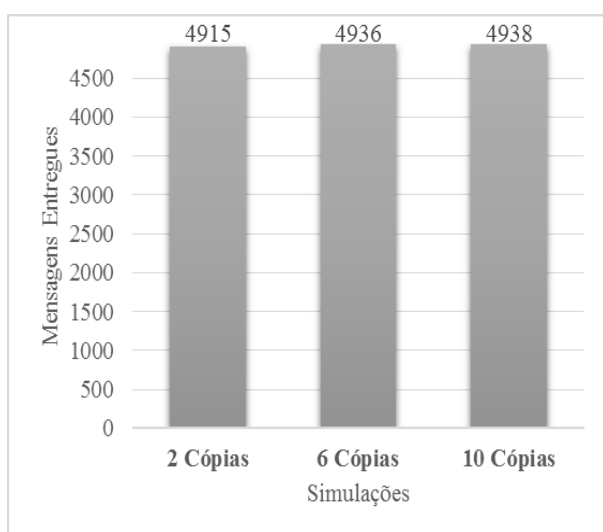


Gráfico 6 - Quantidade de mensagens entregues por simulação.

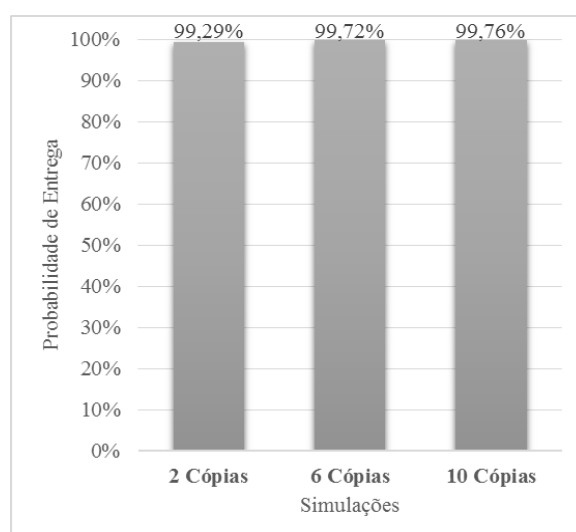


Gráfico 7 - Probabilidade de Entrega da mensagem por simulação.

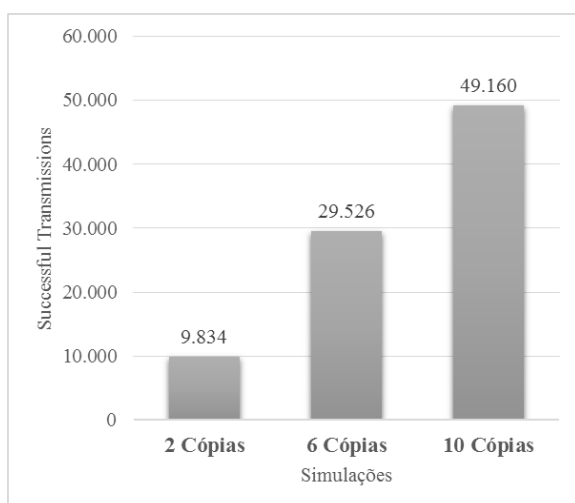


Gráfico 8 - Transmissões bem sucedidas por simulação.

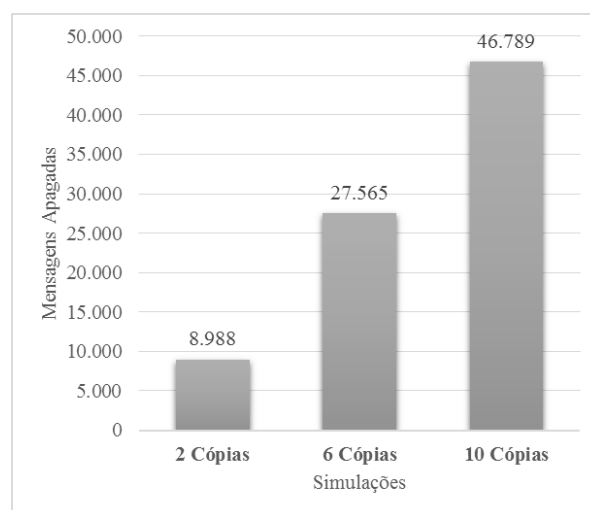


Gráfico 9 - Quantidade de Mensagens Apagadas por simulação.

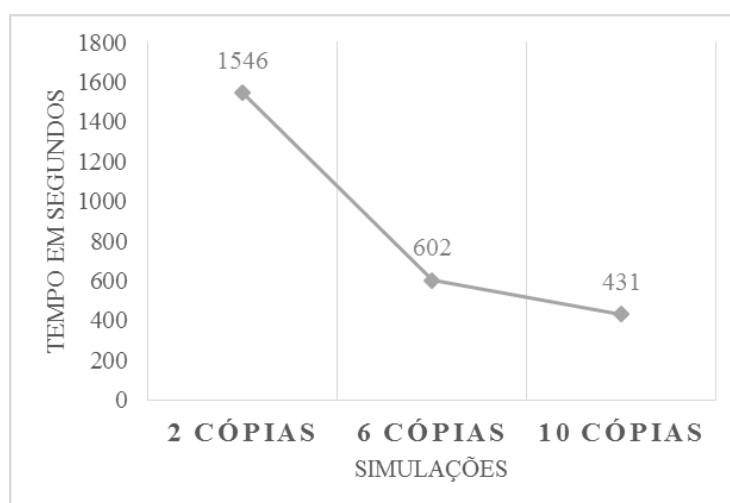


Gráfico 10 - Latência Média em segundos por simulação.

Os gráficos de 11 a 15 foram construídos com base nos relatórios das simulações do cenário de emergência (ou desastre). Assim como nas simulações do segundo cenário, para este cenário não foi realizada uma segunda rodada de simulações, pois buscou-se mesclar a alta mobilidade dos veículos e com a baixa mobilidade dos pedestres. Apesar disso, observa-se que os resultados dos gráficos apresentam uma proximidade com os resultados do cenário veicular.

No caso dos gráficos 11 e 12, onde se apresentam respectivamente a quantidade de mensagens entregues por simulação e a probabilidade de entrega da mensagem por simulação, observasse que o algoritmo obteve um excelente desempenho. No pior caso, que foi a simulação com apenas 2 cópias da mensagem, quase 90% das mensagens foram entregues, ou

seja, das 4950 mensagens criadas, 507 mensagens não foram entregues. Nos outros dois casos, a probabilidade de entrega ficou acima dos 96%, com uma pequena diferença de menos de 1% entre a simulação com 6 cópias e a simulação com 10 cópias, ou seja, uma diferença de 45 mensagens entregues.

Esta performance do algoritmo para este cenário se deve ao fato de que, embora tenham sido definidos muitos nós com baixa mobilidade, todos os nós, na maioria das vezes, estavam transitando entre locais comuns por causa dos pontos de interesse configurados com suas probabilidades para a simulação. Assim, por se deslocarem repetidamente pelos mesmos locais, permitiu-se o favorecimento da entrega das mensagens gerando, conseqüentemente, um excelente desempenho do algoritmo.

Com relação aos gráficos 13 e 14, onde também se apresentam respectivamente a quantidade transmissões bem sucedidas e por simulação e a quantidade de mensagens apagadas por simulação, os resultados também mostram um crescimento proporcional a medida que o número de cópias da mensagem vai aumentando.

Por fim, o gráfico 15, onde é apresentada a latência média para entrega das mensagens, também mostra um comportamento semelhante aos resultados das simulações dos outros cenários. No pior caso, onde a simulação é feita com duas cópias, a latência média foi de 3737 segundos, ou seja, mais de uma hora; e no melhor caso, onde a simulação é feita com 10 cópias, a latência média foi de 1004 segundos, ou seja, aproximadamente 16 minutos.

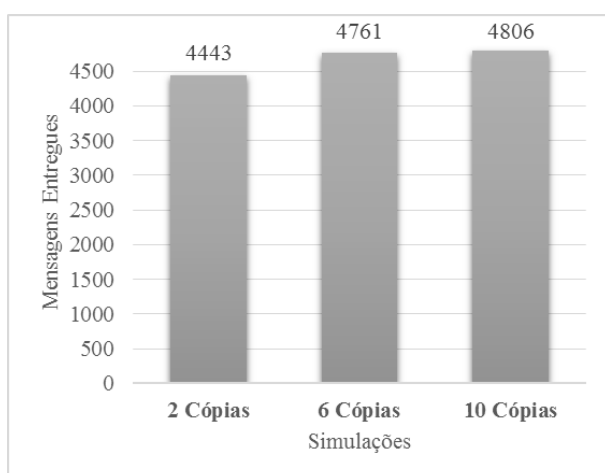


Gráfico 11 - Quantidade de mensagens entregues por simulação.

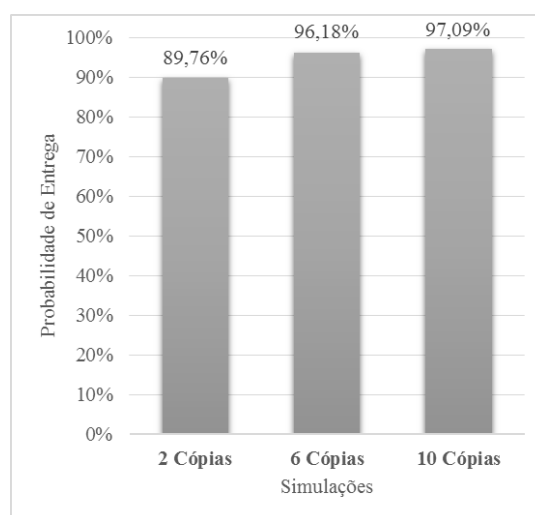


Gráfico 12 - Probabilidade de Entrega da mensagem por simulação.

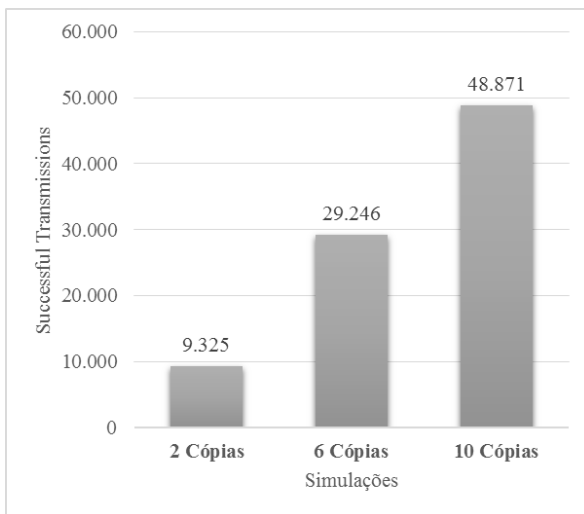


Gráfico 13 - Transmissões bem sucedidas por simulação.

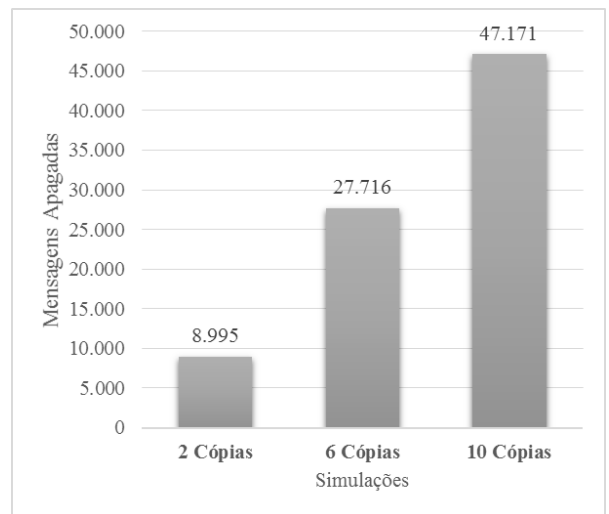


Gráfico 14 - Quantidade de Mensagens Apagadas por simulação

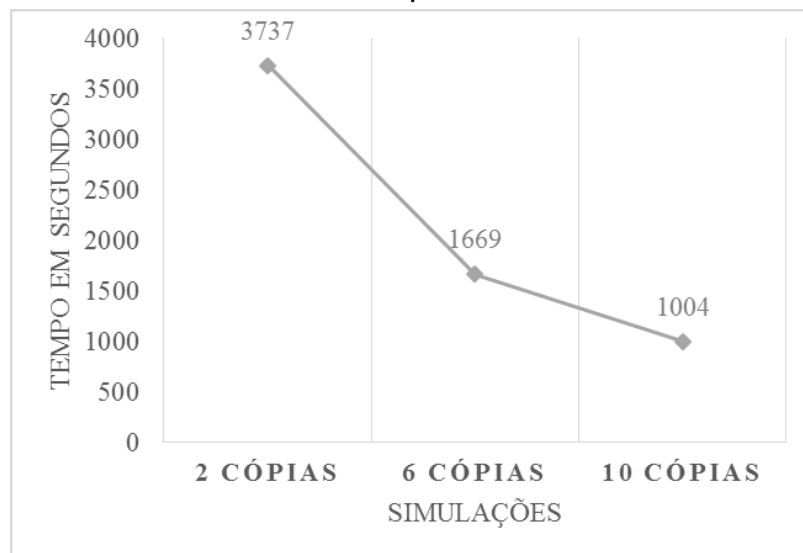


Gráfico 15 - Latência Média em segundos por simulação.

4 CONCLUSÃO

Este trabalho apresentou uma avaliação sobre o algoritmo *Spray and Wait* na comunicação entre dispositivos móveis através de DTNs. Foi realizado um estudo sobre as redes tolerantes a atrasos e desconexões, a qual ainda é uma área emergente de pesquisa; foram estudados os modelos de mobilidade e o algoritmo alvo. Para avaliar o desempenho deste, simulações de cenários urbano, veicular e de emergência foram executadas a fim de se obter os relatórios dos resultados. Estas simulações foram realizadas com o auxílio do simulador The ONE e de outras ferramentas secundárias que foram necessárias para a configuração de cada cenário no simulador. Além disso, buscou-se definir as características de comunicação dos dispositivos móveis entre os nós ao se basear na tecnologia Wi-Fi Direct, que é um tipo de rede sem fio cada vez mais presente nos dispositivos modernos.

É importante salientar que diversos problemas foram enfrentados durante o desenvolvimento deste trabalho, entre eles: a escassa documentação acerca das configurações do simulador e o tempo necessário para executar cada simulação. Para aqueles que forem utilizar o The ONE, recomenda-se que reservem muito tempo para as simulações e utilizem um computador com alto poder de processamento, pois as simulações demoraram muitas horas para serem executadas.

Na seção de análise quantitativa foi possível observar os gráficos construídos com base nos relatórios gerados pelo simulador. Em todos os cenários ficou claro que o principal fator para o algoritmo obter um bom desempenho é a alta mobilidade dos nós. Além disso, observou-se que a quantidade de cópias da mensagem na rede deve ser levada em conta ao se projetar um sistema de comunicação ou monitoramento, pois quanto maior o número de cópias, maior será a demanda de recursos dos nós.

4.1 TRABALHOS FUTUROS

Como o estudo sobre as DTNs ainda é recente e novas tecnologias surgem a cada dia, fica aberta uma porta de possibilidades de estudo e aplicações nos mais diversos cenários. Entretanto, para ser mais específico, um trabalho futuro que pode ser executado é um estudo e desenvolvimento ou adaptação de um algoritmo que tenha como base o *Spray and Wait*, mas

que tenha como proposta específica a possibilidade de enviar uma mensagem para não só um nó específico, mas também para qualquer nó pertencente a um grupo.

Um exemplo de aplicação se daria por utilizar esse algoritmo num cenário de emergência, onde não só as equipes de busca e resgate seriam os nós, mas qualquer pessoa, como um cidadão comum, poderia servir como um nó criador ou transmissor de dados. Desta maneira, caso uma pessoa precisasse de socorro e tivesse um dispositivo que pudesse enviar mensagens de socorro; ou qualquer outro tipo de informação, como uma imagem, um áudio, um vídeo, ou até mesmo a sua geolocalização específica; seria de muita ajuda para equipes de emergência, pois aumentaria a eficiência dos seus trabalhos e, conseqüentemente, resgataria mais vítimas em um menor espaço de tempo.

O uso do algoritmo adaptado do *Spray and Wait* neste caso é apropriado, pois como foi visto nos resultados das simulações, ele possui uma probabilidade de entrega da mensagem muito boa, mesmo quando a quantidade de cópias da mensagem é pequena, além de manter um bom desempenho na análise das outras métricas. Além disso, seria interessante também avaliar o gasto de energia que os nós teriam, que neste caso, se refere ao dispositivo que a pessoa possui, pois numa situação de desastre natural, na maioria das vezes, é cessado o fornecimento de energia.

Um outro trabalho que pode ser feito é a execução deste mesmo estudo sobre a comunicação dos dispositivos móveis através de DTNs, mas desta vez, avaliando o algoritmo proposto por (JUNIOR, 2014) que busca adaptar o conceito de campos magnéticos virtuais com DTN *Anycast*, o qual só permite que exista uma única versão da mensagem na rede.

REFERÊNCIAS

- ALLIANCE, W. **Who We Are**. Disponível em: <<https://www.wi-fi.org/who-we-are>> [Acessado em julho de 2014].
- ALLIANCE, W. **Discover Wi-Fi – Wi-Fi Direct**. Disponível em: <<http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>> [Acessado em setembro de 2014].
- ALMEIDA, V. D. D. **Análise de desempenho de protocolos de roteamento Ad hoc e DTN em redes de emergência**. Universidade Federal de Minas Gerais — Belo Horizonte, 2011
- BRANCO, G. R. et al., **Emprego de redes tolerantes a atrasos e desconexões em cenários de emergência**. Instituto Militar de Engenharia – Rio de Janeiro, 2010.
- CRUZ, I. F.; ASSAKURA, R. **Emprego de redes tolerantes a atrasos e desconexões em cenários de emergência**. Instituto Militar de Engenharia – Rio de Janeiro, 2010.
- DUARTE, G. L. **Redes Tolerantes a Atrasos e Desconexões**. Universidade de São Paulo — São Paulo, 2008
- FERNANDES, R. M. S. **Proposta de um mecanismo de segurança baseado em troca de chaves assimétricas para redes tolerantes a atrasos e desconexões**. UFRJ/COPPE – Rio de Janeiro, 2009.
- GOLASH, S. **Initial available metrics in ONE and explanations of report output**. Disponível em: < http://www.san.uri.br/~ober/arquivos/disciplinas/redes_oportunistas/ONE_output_explained.pdf> [Acessado em fevereiro de 2015].
- JOB, D. H. **Arquitetura para a camada de agregação em redes tolerantes a atrasos e desconexões**. Ministério da Defesa, Exército Brasileiro, Departamento de Ciência e Tecnologia, Instituto Militar de Engenharia — Rio de Janeiro, 2010
- JUANG, P. et al. **Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet**. Princeton University, 2002.
- JUMP, O. **What is OpenJUMP**. Disponível em: < <http://www.openjump.org/>> [Acessado em fevereiro de 15].
- JUNIOR, S. T. D. **Análise de Tráfego De Dados em Redes Bluetooth**. Universidade Federal de Pernambuco – UFPE, 2001.
- JUNIOR, J. A. O.; JUNIOR, L. L.; ANDRADE, G. E.; GISANE, M. **Anycasting in DTNs using Virtual Magnetic Fields**. In: The 11th Annual IEEE Consumer Communications & Networking Conference, 2014, pp. 909.
- KERANEN, A.; OTT, J. **Increasing Reality for DTN Protocol Simulations**. Helsinki University of Technology, 2007.

- KERANEN, A.; OTT, J.; KÄRKKÄINEN, T. **The ONE Simulator for DTN Protocol Evaluation**. Helsinki University of Technology, 2009.
- MEDEIROS, M. V. B.; SALLES, R. M. **Emprego de Redes Tolerantes a Atrasos e Desconexões em Sistemas de Comunicações Militares**. Instituto Militar De Engenharia – Rio de Janeiro, 2009.
- MELO, M. C. L. **Implementação e avaliação de um modelo para troca de mensagens em DTNs utilizando tráfego aéreo**. UFRJ/COPPE – Rio de Janeiro, 2011.
- MOSCHETTO, D. A. et al. **Influência da topologia da área de movimentação na capacidade de transmissão de redes DTN de pessoas**. Anais – XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2011.
- NUNES, C. M. **Roteamento Utilizando Potencial de Entrega em Redes Tolerantes a Atrasos**. Pontifícia Universidade Católica do Rio Grande Do Sul – Porto Alegre, 2010.
- OLIVEIRA, C. T. de et al. **Redes Tolerantes a Atrasos e Desconexões**. Minicursos: 25º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, SBRC'2007, Belém – PA, 54p, Brasil.
- OLIVEIRA, C. T. de. **Uma Proposta de Roteamento Probabilístico para Redes Tolerantes a Atrasos e Desconexões**. COPPE/UFRJ— Rio de Janeiro, 2008
- RASHID, S. et al. **Impact of Mobility Models on DLA (Drop Largest) Optimized DTN Epidemic routing protocol**. Universiti Teknologi Malaysia (UTM) Skudai – Malaysia, 2011.
- SPYROPOULOS, T., PSOUNIS, K.; RAGHAVENDRA, C. **Spray and wait: an efficient routing scheme for intermittently connected mobile networks**. In WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, 2005.
- SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. **Efficient routing in intermittently connected mobile networks: the multiple-copy case**. IEEE/ACM Transactions on Networking, 2008, 16(1):77–90.
- TABARELLI, A.; Silva, C. C. **Redes Tolerantes a Atrasos, Protocolos de Disseminação e Aplicações**. Universidade de São Paulo – São Paulo, 2009.
- TKK/COMNET. **Project page of the ONE simulator**. Disponível em: <<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>> [Acessado em fevereiro de 2015].

GLOSSÁRIO

BRTs – *Bus Rapid Transit*

CONDOR – *Command and Control on the Move Network, Digital Over the Horizon Relay*

DTN – *Delay Tolerant Networks*

IP – *Internet Protocol*

MANET – *Mobile Ad hoc Networks*

ONE – *Opportunistic Network Environment*

PDA's – *Personal Digital Assistants*

PRoPHET – *Probabilistic Routing Protocol using History of Encounters and Transitivity*

POI – *Points of interest*

SeNDT – *Sensor Networking with Delay Tolerance*

SISTAC/Bda – *Sistema Tático de Comunicações de Brigada*

QoS – *Quality of Service*

SPMBM – *Shortest Path Map-Based Movement*

TCP/IP – *Transmission Control Protocol – Internet Protocol*

WKT – *Well-Known Text*

WLAN – *Wireless Local Area Networks*

WiMax – *Worldwide Interoperability for Microwave Access*

APÊNDICE

APÊNDICE A – Script de configuração do cenário urbano com alta mobilidade.

```

Scenario.name = urbano_alta_mobilidade
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
Scenario.endTime = 198000

btInterfaceCarros.type = SimpleBroadcastInterface
btInterfaceCarros.transmitSpeed = 11M
btInterfaceCarros.transmitRange = 95

btInterfacePedestres.type = SimpleBroadcastInterface
btInterfacePedestres.transmitSpeed = 11M
btInterfacePedestres.transmitRange = 35

highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

Scenario.nrofHostGroups = 2

Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
Group.nrofInterfaces = 1
Group.interface1 = btInterfaceCarros
Group.speed = 0.5, 1.5
Group.msgTtl = 300

Group.nrofHosts = 5

Group1.groupID = pedestres
Group1.nrofHosts = 250
Group1.speed = 0.4, 1.0
Group1.interface1 = btInterfacePedestres

Group2.groupID = carros
Group2.nrofHosts = 50
Group2.nrofInterfaces = 2
Group2.interface1 = btInterfacePedestres
Group2.interface2 = btInterfaceCarros
Group2.okMaps = 1
Group2.speed = 2.4, 10

Events.nrof = 1
Events1.class = MessageEventGenerator
Events1.interval = 40
Events1.size = 500k, 1M
Events1.hosts = 0, 299
Events1.prefix = M

MovementModel.rngSeed = 1
MovementModel.worldSize = 4500, 3400
MovementModel.warmup = 1000

```

```

MapBasedMovement.nrofMapFiles = 4

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

Report.nrofReports = 3
Report.warmup = 0
Report.reportDir = reports/
Report.report1 = MessageStatsReport
Report.report2 = ContactsPerHourReport
Report.report3 = MessageDeliveryReport

ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

GUI.UnderlayImage.fileName = data/helsinki_underlay.png
GUI.UnderlayImage.offset = 64, 20
GUI.UnderlayImage.scale = 4.75
GUI.UnderlayImage.rotate = -0.015

GUI.EventLogPanel.nrofEvents = 100

```

APÊNDICE B – Script de configuração do cenário urbano com baixa mobilidade.

```

Scenario.name = urbano_baixa_mobilidade
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
Scenario.endTime = 198000

btInterfaceCarros.type = SimpleBroadcastInterface
btInterfaceCarros.transmitSpeed = 11M
btInterfaceCarros.transmitRange = 95

btInterfacePedestres.type = SimpleBroadcastInterface
btInterfacePedestres.transmitSpeed = 11M
btInterfacePedestres.transmitRange = 35

highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

Scenario.nrofHostGroups = 4

Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
Group.nrofInterfaces = 1
Group.interfacel = btInterfaceCarros
Group.speed = 0.5, 1.5
Group.msgTtl = 300

```

```

Group.nrofHosts = 5

Group1.groupID = pesdesAltaMobi
Group1.nrofHosts = 125
Group1.speed = 0.4, 1.0
Group1.interface1 = btInterfacePedestres

Group2.groupID = pesdesBaixaMobi
Group2.waitTime = 0, 14400
Group2.nrofHosts = 125
Group2.speed = 0.4, 1.0
Group2.interface1 = btInterfacePedestres

Group3.groupID = carrosAltaMobi
Group3.nrofHosts = 25
Group3.nrofInterfaces = 2
Group3.interface1 = btInterfacePedestres
Group3.interface2 = btInterfaceCarros
Group3.okMaps = 1
Group3.speed = 2.4, 10

Group4.groupID = carrosBaixaMobi
Group4.nrofHosts = 25
Group4.waitTime = 0, 14400
Group4.nrofInterfaces = 2
Group4.interface1 = btInterfacePedestres
Group4.interface2 = btInterfaceCarros
Group4.okMaps = 1
Group4.speed = 2.4, 10

Events.nrof = 1
Events1.class = MessageEventGenerator
Events1.interval = 40
Events1.size = 500k,1M
Events1.hosts = 0,299
Events1.prefix = M

MovementModel.rngSeed = 1
MovementModel.worldSize = 4500, 3400
MovementModel.warmup = 1000

MapBasedMovement.nrofMapFiles = 4

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

Report.nrofReports = 3
Report.warmup = 0
Report.reportDir = reports/
Report.report1 = MessageStatsReport
Report.report2 = ContactsPerHourReport
Report.report3 = MessageDeliveryReport

ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

```

```

Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

GUI.UnderlayImage.fileName = data/helsinki_underlay.png
GUI.UnderlayImage.offset = 64, 20
GUI.UnderlayImage.scale = 4.75
GUI.UnderlayImage.rotate = -0.015

GUI.EventLogPanel.nrofEvents = 100

```

APÊNDICE C – Script de configuração do cenário veicular.

```

Scenario.name = cenário_veicular
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
Scenario.endTime = 198000

btInterfaceVeiculos.type = SimpleBroadcastInterface
btInterfaceVeiculos.transmitSpeed = 11M
btInterfaceVeiculos.transmitRange = 95

btInterfacePedestres.type = SimpleBroadcastInterface
btInterfacePedestres.transmitSpeed = 11M
btInterfacePedestres.transmitRange = 35

highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

Scenario.nrofHostGroups = 7

Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
Group.nrofInterfaces = 1
Group.interface1 = btInterfaceVeiculos
Group.speed = 0.5, 1.5
Group.msgTtl = 300
Group.nrofHosts = 5

Group1.groupID = carros
Group1.nrofHosts = 100
Group1.nrofInterfaces = 1
Group1.interface1 = btInterfaceVeiculos
Group1.okMaps = 1
Group1.speed = 2.4, 20

Group2.groupID = caminhos
Group2.nrofHosts = 50
Group2.nrofInterfaces = 1
Group2.interface1 = btInterfaceVeiculos
Group2.okMaps = 1
Group2.speed = 2.4, 14

Group3.groupID = onibus
Group3.nrofHosts = 44
Group3.nrofInterfaces = 1

```

```

Group3.interface1 = btInterfaceVeiculos
Group3.bufferSize = 50M
Group3.movementModel = MapRouteMovement
Group3.routeFile = data/demo_bus.wkt
Group3.routeType = 2
Group3.waitTime = 10, 30
Group3.speed = 2.4, 16

```

```

Group4.groupID = motos
Group4.nrofHosts = 100
Group4.nrofInterfaces = 1
Group4.interface1 = btInterfaceVeiculos
Group4.okMaps = 1
Group4.speed = 2.4, 20

```

```

Group5.groupID = bondinhoAzul
Group5.bufferSize = 50M
Group5.movementModel = MapRouteMovement
Group5.routeFile = data/tram3.wkt
Group5.routeType = 1
Group5.waitTime = 10, 30
Group5.speed = 7, 10
Group5.nrofHosts = 2
Group5.nrofInterfaces = 1
Group5.interface1 = btInterfaceVeiculos

```

```

Group6.groupID = bondinhoVerde
Group6.bufferSize = 50M
Group6.movementModel = MapRouteMovement
Group6.routeFile = data/tram4.wkt
Group6.routeType = 2
Group6.waitTime = 10, 30
Group6.speed = 7, 10
Group6.nrofHosts = 2
Group6.nrofInterfaces = 1
Group6.interface1 = btInterfaceVeiculos

```

```

Group7.groupID = bondinhoBranco
Group7.bufferSize = 50M
Group7.movementModel = MapRouteMovement
Group7.routeFile = data/tram10.wkt
Group7.routeType = 2
Group7.waitTime = 10, 30
Group7.speed = 7, 10
Group7.nrofHosts = 2
Group7.nrofInterfaces = 1
Group7.interface1 = btInterfaceVeiculos

```

```

Events.nrof = 1
Events1.class = MessageEventGenerator
Events1.interval = 40
Events1.size = 500k,1M
Events1.hosts = 0,299
Events1.prefix = M

```

```

MovementModel.rngSeed = 1
MovementModel.worldSize = 4500, 3400
MovementModel.warmup = 1000

```

```

MapBasedMovement.nrofMapFiles = 4

```

```

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/demo_bus.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

Report.nrofReports = 3
Report.warmup = 0
Report.reportDir = reports/
Report.report1 = MessageStatsReport
Report.report2 = ContactsPerHourReport
Report.report3 = MessageDeliveryReport

ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

GUI.UnderlayImage.offset = 64, 20
GUI.UnderlayImage.scale = 4.75
GUI.UnderlayImage.rotate = -0.015

GUI.EventLogPanel.nrofEvents = 100
RE-matching details)

```

APÊNDICE D – Script de configuração do cenário de emergência (ou desastre).

```

Scenario.name = cenario_desastre
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
Scenario.endTime = 198000

btInterfaceCarros.type = SimpleBroadcastInterface
btInterfaceCarros.transmitSpeed = 11M
btInterfaceCarros.transmitRange = 95

btInterfaceVeiculos.type = SimpleBroadcastInterface
btInterfaceVeiculos.transmitSpeed = 11M
btInterfaceVeiculos.transmitRange = 35

highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

Scenario.nrofHostGroups = 8

Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
Group.nrofInterfaces = 1
Group.interface1 = btInterfaceVeiculos
Group.speed = 0.5, 1.5
Group.msgTtl = 300

Group.nrofHosts = 1

```



```

Group1.groupID = hospital1
Group1.movementModel = StationaryMovement
Group1.nodeLocation = 514, 1521

```

```

Group2.groupID = centroDEcontrole
Group2.movementModel = StationaryMovement
Group2.nodeLocation = 667,1796

```

```

Group3.groupID = hospital2
Group3.movementModel = StationaryMovement
Group3.nodeLocation = 935, 1690

```

```

Group4.groupID = carros
Group4.nrofHosts = 30
Group4.nrofInterfaces = 1
Group4.interface1 = btInterfaceVeiculos
Group4.interface2 = btInterfaceVeiculos
Group4.okMaps = 1
Group4.speed = 2.4, 20

```

```

Group5.groupID = caminhoes
Group5.nrofHosts = 10
Group5.nrofInterfaces = 1
Group5.interface1 = btInterfaceVeiculos
Group5.interface2 = btInterfaceVeiculos
Group5.okMaps = 1
Group5.speed = 2.4, 14

```

```

Group6.groupID = equipesPedesAltaMobi
Group6.nrofHosts = 120
Group6.speed = 0.4, 1.0
Group6.interface1 = btInterfaceVeiculos

```

```

Group7.groupID = equipesPedesBaixaMobi
Group7.waitTime = 0, 14400
Group7.nrofHosts = 107
Group7.speed = 0.4, 1.0
Group7.interface1 = btInterfaceVeiculos

```

```

Group8.groupID = motos
Group8.nrofHosts = 30
Group8.nrofInterfaces = 1
Group8.interface1 = btInterfaceVeiculos
Group8.interface2 = btInterfaceVeiculos
Group8.okMaps = 1
Group8.speed = 2.4, 20

```

```

Events.nrof = 1
Events1.class = MessageEventGenerator
Events1.interval = 40
Events1.size = 500k,1M
Events1.hosts = 0,229
Events1.prefix = M

```

```

MovementModel.rngSeed = 1
MovementModel.worldSize = 4500, 3400
MovementModel.warmup = 1000

```

```

MapBasedMovement.nrofMapFiles = 4

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

PointsOfInterest.poiFile1 = data/WestPOIs.wkt
PointsOfInterest.poiFile2 = data/CentralPOIs.wkt
PointsOfInterest.poiFile3 = data/EscolaPOIs.wkt
PointsOfInterest.poiFile4 = data/EstacaoTransbordoPOIs.wkt

Group4.pois = 1,0.4, 2,0.2, 3,0.2, 4, 0.2
Group5.pois = 1,0.4, 2,0.2, 3,0.2, 4, 0.2
Group6.pois = 1,0.4, 2,0.2, 3,0.2, 4, 0.2
Group7.pois = 1,0.4, 2,0.2, 3,0.2, 4, 0.2
Group8.pois = 1,0.4, 2,0.2, 3,0.2, 4, 0.2

Report.nrofReports = 1
Report.warmup = 0
Report.reportDir = reports/
Report.report1 = MessageStatsReport

ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

GUI.UnderlayImage.fileName = data/helsinki_underlay.png
GUI.UnderlayImage.offset = 64, 20
GUI.UnderlayImage.scale = 4.75
GUI.UnderlayImage.rotate = -0.015

GUI.EventLogPanel.nrofEvents = 100

```

APÊNDICE E – Script de configuração do algoritmo Spray and Wait.

```

Scenario.name = SprayAndWait
Group.router = SprayAndWaitRouter
SprayAndWaitRouter.nrofCopies = 2
#SprayAndWaitRouter.nrofCopies = [2;6;10]
SprayAndWaitRouter.binaryMode = true

```